English

Fujitsu Software BS2000

VM2000

Virtual Machine System

User Guide

Valid for:
VM2000 V12.0B

Edition November 2023

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: bs2000services@fujitsu.com.

Certified documentation according to DIN EN ISO 9001:2015

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 9001:2015.

Copyright and Trademarks

Copyright © 2023 Fujitsu Technology Solutions GmbH.

All rights reserved.
Delivery subject to availability; right of technical modifications reserved.

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

The FUJITSU trademark and the FUJITSU logo are registered trademarks of Fujitsu Limited, Japan, in Europe and in other countries.

BS2000 is a brand of Fujitsu Technology Solutions GmbH in Europe.

ETERNUS is a trademark of Fujitsu Limited, Japan, in Europe and in other countries.

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.

Table of Contents

VM2000 (EN) 10
1 Preface 11
1.1 Objectives and target groups of this manual 12
1.2 Summary of contents 13
1.3 Changes since the last edition of the manual 14
1.4 Notational conventions 16
1.5 Possible areas of application for VM2000 17
1.6 VM2000 characteristics 19
2 Structure of VM2000 20
2.1 Architecture of the SE servers 21
2.2 Terms of VM2000 23
2.3 Implementation of VM2000 26
2.3.1 Implementation on SU /390 27
2.3.2 Implementation on SU x86 30
2.4 Resource management under VM2000 34
2.5 Supported hardware and operating systems 35
2.6 Constraints 36
2.6.1 Constraints that apply to the hardware 37
2.6.2 Constraints that apply to VM2000 38
2.7 Comparison of VM2000 on SU /390 and SU x86 39
3 Installing VM2000 41
3.1 Generating and configuring the device peripherals 43
3.1.1 Device peripherals on SU /390 44
3.1.2 BS2000 devices connected to SU x86 46
3.2 Setting up the VM2000 monitor 48
3.3 Installing the VM2000 message files 51
3.4 Installing the VM2000 hypervisor files (SU /390) 52
3.5 Setting the VM2000 parameters in the parameter service 53
3.6 Configuration settings for the monitor VM 55
3.6.1 Configuration via the VM2000 configuration file (SU /390) 56
3.6.2 Configuration using the SE Manager (SU x86) 59
3.7 Preparing to use $VMCONS 60
3.8 Installing the guest systems 61
4 Initializing and terminating VM2000 62
4.1 Initializing VM2000 operation 63
4.1.1 Operating modes 64
4.1.2 Starting the monitor system 65

4.2 Starting up the monitor system 66
4.2.1 Required components 67
4.2.2 Startup procedure in the monitor system 68
4.2.3 Example of startup in the monitor system 71
4.3 Terminating the monitor system and VM2000 77
4.3.1 Coordinated shutdown of VM2000 79
4.3.2 Shutdown in the monitor system 82
4.4 Loading, operating and terminating guest systems 83
5 Operating VM2000 and the virtual machines (VMs) 84
5.1 VM2000 Operating strategy 85
5.1.1 Operating levels of VM2000 86
5.1.2 Operating the virtual machines (ADMIN dialog from user task) 88
5.1.2.1 Privileges of the user task in the monitor system 89
5.1.2.2 Administration dialogs from privileged user tasks 90
5.1.2.3 Example of an ADMIN dialog from a user task 92
5.1.3 Operating the guest systems using BS2000 consoles 96
5.1.3.1 Operating the guest systems using KVP consoles 97
5.1.3.2 Operating the guest systems using logical consoles 99
5.1.4 VM2000 administration application $VMCONS 100
5.1.4.1 Establishing a connection to $VMCONS 102
5.1.4.2 Operating the virtual machines (ADMIN dialog via $VMCONS) 105
5.1.4.3 Operating the guest systems (VC dialog via $VMCONS) 107
5.1.4.4 Example of ADMIN and VC dialog via $VMCONS 112
5.1.5 Automating VM2000 and VM administration 117
5.2 Operating a VM 122
5.2.1 Initializing a VM 124
5.2.2 Monitoring a VM with job variables (MONJV) 132
5.2.3 Managing devices and assignment sets 138
5.2.4 VM and VM group 139
5.2.5 Assigning a VM or a VM group to a CPU pool 140
5.2.6 Starting VM administration 141
5.2.7 Starting the guest system on a VM 142
5.2.8 Information services 145
5.2.9 Resource management 147
5.2.10 Suspending a VM / resuming operation 148
5.2.11 Moving a VM within main memory 151
5.2.12 Sends a message to VM administrators 152
5.2.13 Modifying the VMCONS parameters 153
5.2.14 Terminating VM administration 154
5.2.15 Aborting the guest system on a VM 155
5.2.16 Shutdown in the guest system and of VM2000 156

5.2.17 Terminating a VM 159
5.3 VM Migration 161
5.3.1 Prerequisites for the VM Migration 163
5.3.2 Phases of a VM Migration 165
5.3.3 CPU management during a VM Migration 167
5.3.4 Device configuration for the VM Migration (SU /390) 168
5.3.5 Device management during a VM Migration 170
5.3.6 Relocation of the VM definition during VM Migration 171
5.3.7 Commands not allowed during a VM Migration 172
5.3.8 Monitoring a VM Migration with job variables (MONJV) 173
5.3.9 Example for a VM Migration (SU /390) 176
5.4 Working with VM definitions 186
5.4.1 VM definition and previous VM2000 commands 188
5.4.2 Commands for managing VM definitions 189
5.4.3 Examples of working with VM definitions 193
5.5 Operating VM groups (SU /390) 201
5.5.1 Creating a VM group 203
5.5.2 Adding a VM to a VM group 204
5.5.3 Modifying the attributes of a VM in a VM group 205
5.5.4 Removing a VM from a VM group 206
5.5.5 Modifying the attributes of a VM group 207
5.5.6 Outputting information on a VM group 208
5.5.7 Deleting a VM group 209
5.5.8 Example of operating VM groups 210
5.6 Operating the whole VM2000 system 213
5.6.1 Types of command 214
5.6.2 Overview of commands for VM2000 operation 215
5.6.3 Privileged functions 218
5.6.4 Restrictions in VM2000 commands/functions 219
5.6.5 Extending and restricting the range of commands/functions 220
5.6.6 VM2000 functions in the SE Manager 222
5.7 VM2000 in the guest systems 224
5.7.1 ATTACH-DEVICE (Attach hardware units) 226
5.7.2 DETACH-DEVICE (Detach hardware units) 228
5.7.3 INCLUDE-/REMOVE-DEVICE-CONNECTION (Add/Remove logical connections) 230
5.7.4 MODIFY-IO-UNIT (Modify configuration properties of an I/O unit) 231
5.7.5 MODIFY-IO-OPTIONS (Modify settings for I/O devices) 232
5.7.6 SHOW-SYSTEM-INFORMATION (Display system information) 234
5.7.7 SHOW-DEVICE-CONFIGURATION (Display device configuration and availability) 239

6 VM2000 management 243
6.1 CPU management 244
6.1.1 Managing and reconfiguring real and virtual CPUs 245
6.1.1.1 Real CPUs 246
6.1.1.2 Virtual CPUs 248
6.1.1.3 Reconfiguration of real and virtual CPUs 251
6.1.1.4 High availability with hot spare CPUs (SU /390) 253
6.1.1.5 Increasing capacity with extra CPUs 255
6.1.2 Managing CPU pools 258
6.1.2.1 Static CPU pools 259
6.1.2.2 Dynamic CPU pools 260
6.1.2.3 Number of attached real and virtual CPUs in the CPU pool 262
6.1.2.4 Creating a CPU pool 265
6.1.2.5 Adding CPUs to a CPU pool 266
6.1.2.6 Assigning a VM or VM group to a (different) CPU pool 267
6.1.2.7 Canceling the assignment of a VM or VM group to a CPU pool 268
6.1.2.8 Outputting information on a CPU pool 269
6.1.2.9 Deleting a CPU pool 270
6.1.2.10 Example of managing CPU pools 271
6.1.3 Planning distribution of the CPU capacity to the VMs 275
6.1.4 Scheduling procedure 286
6.1.4.1 Fixed CPU assignment (dedicated CPUs, SU /390) 287
6.1.4.2 CPU assignment in time slicing (SU /390) 289
6.1.5 Example of CPU management 291
6.2 Reconfiguring Channels, controllers and paths (SU /390) 298
6.3 Main memory management and reconfiguration 300
6.3.1 Managing main memory 301
6.3.1.1 Managing main memory on SU /390 302
6.3.1.2 Managing main memory on SU x86 305
6.3.2 Reconfiguring main memory 306
6.4 Peripheral device management 312
6.4.1 Assigning and removing devices of a VM 313
6.4.1.1 Explicit device assignment and removal 314
6.4.1.2 Assignment sets, implicit device assignment and release 315
6.4.1.3 Interplay between implicit and explicit device assignment 318
6.4.1.4 Usage mode of a device 319
6.4.1.5 Example of peripheral device management 321
6.4.2 Assigning and removing pubsets of a VM 331
6.4.2.1 Explicit pubset assignment and removal 333
6.4.2.2 Implicit pubset assignment and release 335
6.4.2.3 Interplay between implicit and explicit pubset assignment 337

6.4.2.4 Usage mode of pubsets . . . 338
6.4.2.5 Notes on pubset specification in the VM2000 commands . . . 339
6.4.2.6 Pubset reconfiguration . . . 341
6.4.2.7 Mirror pubsets . . . 343
6.4.2.8 Example of pubset management . . . 345
6.4.3 Special devices . . . 356
6.4.4 Shared disk devices . . . 359
6.4.5 Parallel access volume (PAV, SU /390) . . . 363
6.4.6 Using the Highspeed Net Connect HNC (SU /390) . . . 370
6.4.7 Notes on configuration for VM2000 operation . . . 373
6.4.8 Use of IORM in VM2000 operation . . . 374
6.4.9 Dynamic I/O configuration modification . . . 376
6.4.9.1 Changing the configuration of channel devices using BS2000 (SU /390) . 377
6.4.9.2 Changing the configuration of devices via X2000 (SU x86) . . . 378
6.4.9.3 Changing the configuration of devices in the disk storage system . . . 379
6.4.10 Checking and updating of the VM2000 disk configuration (SU /390) . . . 380
6.5 Backing up and restoring the VM2000 configuration . . . 381
6.5.1 Backing up a VM2000 configuration . . . 382
6.5.2 Modifying the backup of a VM2000 configuration . . . 383
6.5.3 Restoring a VM2000 configuration . . . 384
6.5.4 Backup file for a VM2000 configuration . . . 386
6.5.4.1 Storage location . . . 387
6.5.4.2 Structure of the contents . . . 388
6.5.4.3 VM2000 components not backed up . . . 391
6.5.4.4 Example of a backup file (SU /390) . . . 392
6.6 Password protection . . . 395
6.7 Accounting in VM2000 . . . 396
6.7.1 Accounting record for devices . . . 398
6.7.2 Accounting record for a VM . . . 400
6.8 Hardware error logging in VM2000 . . . 401
6.9 Monitoring in VM2000 . . . 404
6.9.1 Measurement with /SHOW-VM-STATUS . . . 405
6.9.2 Measurement with the openSM2 measurement monitor . . . 406
7 VM2000 commands . . . 407
7.1 Overview of commands for VM2000 operation . . . 408
7.1.1 Table of VM2000 commands . . . 409
7.1.2 Grouping of VM2000 commands according to function . . . 412
7.2 Command return codes . . . 416
7.3 Guaranteed messages . . . 418
7.4 SHOW output in S variables . . . 419
7.5 Description of the commands . . . 420

7.5.1 ACTIVATE-VM-DEFINITION (Activate a VM definition) 423
7.5.2 ADD-VM-DEVICES (Allocate devices to a VM) 426
7.5.3 ADD-VM-TO-VM-GROUP (Add a VM to a VM group) 430
7.5.4 ASSIGN-VM-GROUP-TO-CPU-POOL (Assign a VM group to a CPU pool) . 432
7.5.5 ASSIGN-VM-TO-CPU-POOL (Assign a VM to a CPU pool) 434
7.5.6 ATTACH-VM-RESOURCES (Attach real CPUs) 436
7.5.7 BEGIN-VM-DIALOG (Start administration or operation) 441
7.5.8 CALL-VM-PROCEDURE (Start a VM2000 procedure file) 444
7.5.9 CHECK-VM-DISK-CONFIGURATION (Check VM2000 disk configuration) . . 447
7.5.10 CHECK-VM-MIGRATION (Checking prerequisites for VM Migration) 449
7.5.11 CREATE-VM (Initialize a VM) 451
7.5.12 CREATE-VM-ASSIGNMENT-SET (Create assignment set) 461
7.5.13 CREATE-VM-CPU-POOL (Create CPU pool) 462
7.5.14 CREATE-VM-DEFINITION (Create VM definition) 463
7.5.15 CREATE-VM-GROUP (Create VM group) 473
7.5.16 DELETE-VM (Terminate a VM) 475
7.5.17 DELETE-VM-ASSIGNMENT-SET (Delete assignment set) 477
7.5.18 DELETE-VM-CPU-POOL (Delete CPU pool) 478
7.5.19 DELETE-VM-DEFINITION (Delete VM definition) 479
7.5.20 DELETE-VM-GROUP (Delete VM group) 480
7.5.21 DETACH-VM-RESOURCES (Detach real CPUs) 481
7.5.22 END-VM-DIALOG (Terminate administration or operation) 486
7.5.23 EXTEND-VM-MEMORY (Increase main memory for a VM) 488
7.5.24 HOLD-VM (Suspend VM) 492
7.5.25 MIGRATE-VM (Migrate VM) 494
7.5.26 MIGRATE-VM-DEFINITION (Migrate VM definition) 497
7.5.27 MODIFY-VM-ATTRIBUTES (Modify attributes of a VM) 499
7.5.28 MODIFY-VM-DEFINITION (Modify VM definition) 507
7.5.29 MODIFY-VM-DEVICE-ATTRIBUTES (Modify device attributes) 521
7.5.30 MODIFY-VM-DEVICE-USAGE (Modify usage mode of devices of a VM) . . 525
7.5.31 MODIFY-VM-GROUP-ATTRIBUTES (Modify attributes of a VM group) . . . 529
7.5.32 MODIFY-VMCONS-PARAMETERS (Modify parameters of $VMCONS) . . . 531
7.5.33 MOVE-VM (Move a VM within main memory) 533
7.5.34 RECOVER-VM-DEFINITION (Recover a VM definition) 536
7.5.35 REDUCE-VM-MEMORY (Reduce main memory for a VM) 538
7.5.36 REMOVE-VM-DEVICES (Deallocate devices from VM) 541
7.5.37 REMOVE-VM-FROM-VM-GROUP (Remove VM from VM group) 546
7.5.38 RESET-VM (Abort the guest system on a VM) 548
7.5.39 RESTORE-VM-CONFIGURATION (Restoring the VM2000 configuration) . 550
7.5.40 RESUME-VM (Resume VM operation) 552
7.5.41 SAVE-VM-CONFIGURATION (Save the VM2000 configuration) 554

7.5.42 SEND-VM-MESSAGE (Pass messages to VM administrators) 556
7.5.43 SHOW-VM-ASSIGNMENT-SET (Output assignment sets) 558
7.5.44 SHOW-VM-ATTRIBUTES (Output attributes of a VM) 560
7.5.45 SHOW-VM-CPU-POOL (Output attributes of a CPU pool) 576
7.5.46 SHOW-VM-DEFINITION (Output attributes of a VM definition) 583
7.5.47 SHOW-VM-DEVICE-STATUS (Output detailed device information) 596
7.5.48 SHOW-VM-GROUP (Output attributes of VM group) 610
7.5.49 SHOW-VM-RESOURCES (Output status of VM2000 resources) 615
7.5.50 SHOW-VM-STATUS (Output VM2000 monitored data) 639
7.5.51 SHUTDOWN-VM (Execute shutdown in a guest system and for VM2000) . 653
7.5.52 START-VM (Start a guest system on a VM) 657
7.5.53 SWITCH-VM-CPU (Remove CPUs from a CPU pool and add them to another CPU pool) 661
7.5.54 SWITCH-VM-DEVICES (Remove devices from one VM and add them to another) 665
8 Error handling in VM2000 671
8.1 Types of error in VM2000 672
8.1.1 Errors in the carrier system (SU x86) 673
8.1.2 VM2000 errors 674
8.1.3 Guest system errors 677
8.2 Restart routines in VM2000 680
8.2.1 Automatic restart following VM2000 hypervisor termination (SU /390) 681
8.2.2 Automatic restart following termination of the VM2000 Agent 682
8.2.3 Automatic restart after SETS in the monitor system 683
8.2.4 Restarting the monitor system via SVP (SU /390) 684
8.2.5 Restarting a task after an error in a system task of VM2000 685
8.3 VMDUMP utility routine 686
8.4 Other diagnostic aids 688
8.5 Diagnostic file SLEDFILE 689
8.5.1 Creating SLEDFILEs during VM2000 operation 690
8.5.2 Analyzing SLEDFILEs with DAMP 692
9 Glossary 695
10 Related publications 699

1 Preface

Setting up a virtual infrastructure under BS2000 by using the **VM2000 virtual machine system** can lower IT costs by improving efficiency, flexibility and reactivity. It enables you to quickly reallocate IT resources following daily changing demands, provide new services and drastically improve the capacity utilization of the Server Units.

VM2000 permits the simultaneous operation of different, completely self-contained system environments on a single Server Unit with almost the same performance as in native mode (operation without VM2000). VM2000 enhances system versatility and improves the capacity utilization of the Server Unit. If necessary, virtual machines running under VM2000 can be migrated to other Server Units by means of a VM Migration (Live Migration).

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 Server Units and to SU clusters.

VM2000 V12.0 operates the current Fujitsu Server BS2000 SE Series (Server Units /390 and x86) in all capacity levels.

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 SE 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 operating system and knowledge of the operation of the SE server are essential for an understanding of this manual.

The central manual for operating SE servers is the manual “Operation and Administration” [19]. In addition to the concepts and functions for SE servers, this manual also describes in detail how to work with the SE manager, the user interface of the SE servers.

1.2 Summary of contents

VM2000 V12.0 is the evolutionary development of the software product VM2000 V11.5 (SE servers).

When different approaches to a topic exist for the different architectures of the Server Units, each approach is described in a separate section.

The function and command descriptions contain information which only applies for particular Server Units. This information is labeled, see section "Notational conventions".

> ¡ A comparative overview of the differences between the various Server Units is provided in the section "Comparison of VM2000 on SU /390 and SU x86".

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 "VM2000 commands" contains the syntax and semantics of all of the VM2000 commands. It is aimed at VM2000 administrators and VM 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.

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 https://bs2manuals.ts.fujitsu.com.

1.3 Changes since the last edition of the manual

The manual for VM2000 V12.0 describes the use of VM2000 on SE servers.

Compared to the last edition of the manual, VM2000 V11.5, the following important changes have been made:

- VM2000 V12.0 is executable in a monitor system with BS2000 V21.0; guest systems BS2000 OSD/BC V10.0 are no more supported (see "Supported hardware and operating systems").
- The version of BS2000 systems is not output by VM2000 information commands in form of a BS2000/OSD version (e.g. V11.0B) but in form of a BS2000 version (e.g. V20.0B).
- All references to XenVMs, i.e. to Linux and Windows guest systems, were eliminated.
  On SU x86, the carrier system X2000 supported the management of XenVMs via the SE Manager so far. XenVMs were unknown in VM2000 but they were regarded in the display of global resources under the keyword FOREIGN. The management of XenVMs is no longer supported within the current carrier systems, on which VM2000 V12.0 can run.
- Default value *UNIQUE of the parameter SERVER-UNIT-NAME was introduced for VM2000 commands /CHECK-VM-MIGRATION, /MIGRATE-VM and /MIGRATE-VM-DEFINITION.
- On SU x86, VM2000 commands /HOLD-VM and /RESUME-VM are rejected with message VMS3011 and command return code 0|64|VMS9004.
- The obsolete device families *DISKETTE, *EXOTIC and *UNMTAPE cannot be specified any longer.
- Regarding "Parallel Access Volume (PAV, SU /390)", the support of function "FastDPAV" under VM2000 is described.
- In the section "VM2000 in the guest systems", the new BS2000 command /MODIFY-IO-OPTIONS and its effects in the VM2000 operation are described.

Incompatibilities in the VM2000 information commands

There are following incompatible changes in the output of the VM2000 information commands:

- The BS2000 version (instead of the BS2000/OSD version) is output to the SYSOUT and to the S variable GUEST-VERSION with commands /SHOW-VM-ATTRIBUTES and /SHOW-VM-RESOURCES. Furthermore, string „BS2000“ (instead of „BS2-OSD“) is output to the S variable GUEST-TYPE.
- The parameter INFORMATION of commands /SHOW-VM-ATTRIBUTES and /SHOW-VM-RESOURCES does not support the value “*GS” any more.
  The S variables PART-ID, GS-ATTR, GS-MODE, GS-STA, GS-SIZE (VM specific) and GS-SIZE (global configuration) were eliminated entirely.
- The S variables PATH-GROUP-ID-MODE, NUM-OF-VIRT-CPUS and REAL-MIN-MEM-SIZE were eliminated from the output of command /SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION .
- The parameter INFORMATION of commands /SHOW-VM-RESOURCES and /SHOW-VM-DEVICE-STATUS does not support the value “*HA-CLUSTER” any more.

Additional changes in the manual VM2000 V12.0B compared to VM2000 V12.0A

- The main memory of a VM on SU x86 must be at least 1024 Mbytes in size.

- Elimination of Symmetrix disk storage systems (no support any more).
- Updates of command return codes and examples.

1.4 Notational conventions

The following abbreviations are used in this manual:

- **BS2000 servers** as the synonym for SE servers (SU /390 and x86). BS2000 servers are operated with the corresponding BS2000 operating system.
- **SE server** for the Fujitsu Server BS2000 SE Series (Server Units /390 and x86)
- **SU /390** for the Server Unit /390 (/390 architecture)
- **SU x86** for the Server Unit x86 (x86-64 architecture)
- **VM2000** as a synonym for the current version of VM2000

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.

The following notational conventions are used in this manual:

i 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 [13].

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, **`bold 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 Units. This information is labeled as follows:

390 This comment applies only for SU /390

x86 This comment applies only for SU x86

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 SE (Server Units /390 and x86) 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 Unit 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 and main memory.

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 Unit, 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 (SU /390).
- **Creating VM groups (SU /390)**
  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 Unit definitions are possible in the one-digit RPF area up to one percent of the CPU capacity.

VM Migration (Live Migration) under VM2000

To increase the availability and improve the load balancing between the Server Units, it is possible, on a Server Unit /390 or x86, to migrate a virtual machine during ongoing guest system operation under VM2000 from the local Server Unit to another Server Unit of the same type, while containing the VM’s resources.

1.6 VM2000 characteristics

VM2000 is characterized by the following features:

- **Concurrent** execution of up to 32 BS2000 guest systems on a **single** Server Unit.

  Guest systems can be the operating systems released for this VM2000 version (see "Supported hardware and operating systems").

  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 in VM2000.

Further information on the use of VM2000 in terms of performance is provided in the “Performance Handbook” [18].

2 Structure of VM2000

- Architecture of the SE servers
- Terms of VM2000
- Implementation of VM2000
  - Implementation on SU /390
  - Implementation on SU x86
- Resource management under VM2000
- Supported hardware and operating systems
- Constraints
  - Constraints that apply to the hardware
  - Constraints that apply to VM2000
- Comparison of VM2000 on SU /390 and SU x86

2.1 Architecture of the SE servers

A Fujitsu Server BS2000 SE series (for short: SE server) consists of the following components in its maximum configuration:

- Server Units (SU /390 and SU x86)
- Application Units (AU)
- Peripherals (storage)
- Management Unit (MU) with SE Manager
- Net Unit, for SU /390 with HNC

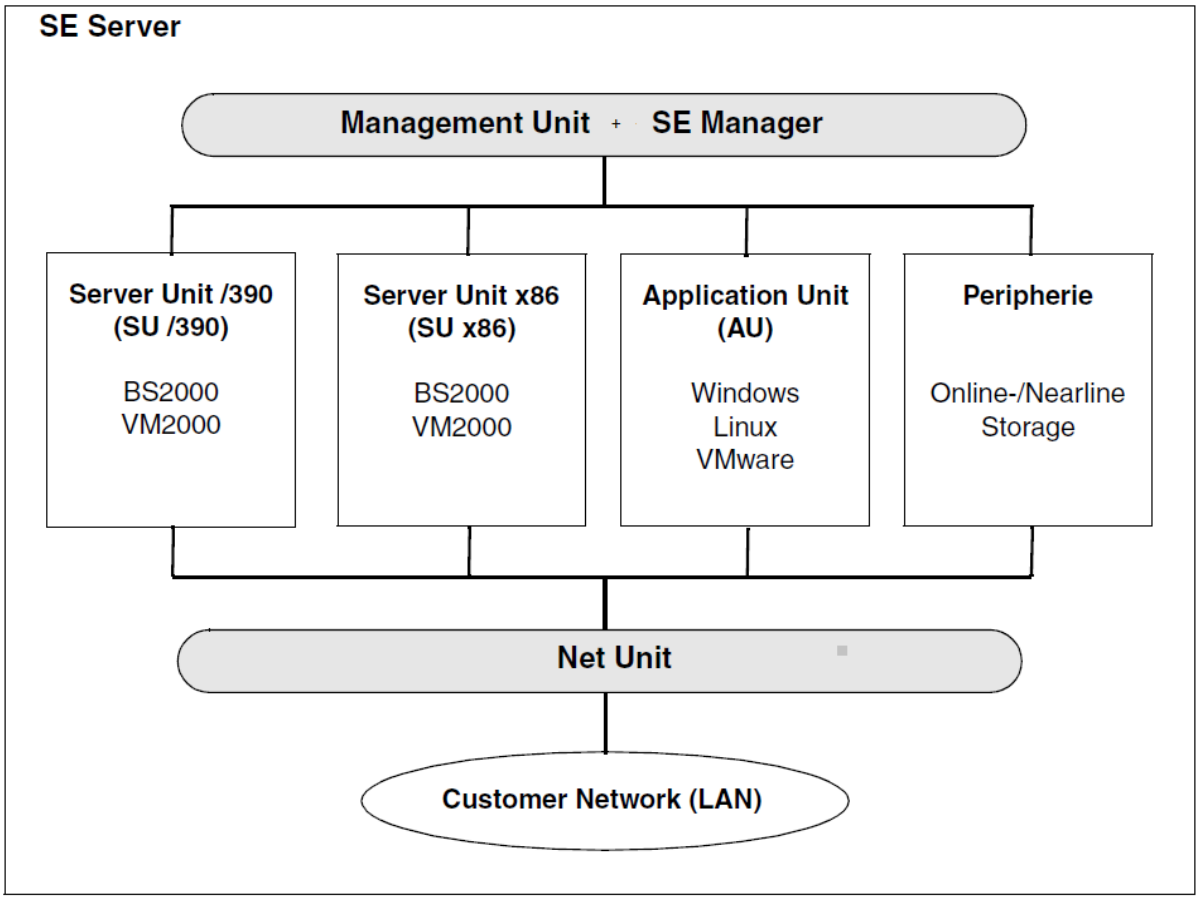


Figure 1: Architecture of the SE servers

With the SE Manager you can operate and manage all components of the SE server centrally from the Management Unit. The SE Manager offers a user-friendly, web-based user interface for this purpose.

In addition to the maximum configuration, variants are also offered which do not contain all Units.

Operation of the SE Manager is described in the online help for the SE Manager and in the “Operation and Administration” manual [19].

The Net Unit, for SU /390 with High-Speed Network Connect (HNC), offers highest performance and security for internal communication in an SE server and for the connection to customer networks (LAN).

The BS2000 operating system in the released versions serves the Server Unit /390 (/390 server) and the Server Unit x86 (x86 server).

> Further configuration levels of the Server Units may be released at some point in the future. Watch out for product announcements and release information.

2.2 Terms of VM2000

VM2000, virtual machine, guest system

The VM2000 virtual machine system is a software product which implements **more than one virtual machine (VM)** on **one** real SE server (more precisely: on a Server Unit of the SE server).

From the user's point of view, a virtual machine is a Server Unit on which his/her operating system, the **guest system**, runs.

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 "Possible areas of application for VM2000".

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".

The administration application $VMCONS (optional, see "VM2000 administration application $VMCONS") 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 "Assignment sets, implicit device assignment and release").

These interfaces are described in detail in section "VM2000 in the guest systems".

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).

390 On SU /390, the VM2000 hypervisor is a separate load module of VM2000 which is loaded (automatically) when VM2000 operation is initialized.

x86 On SU x86, 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 Server Unit offers for an operating system is copied and made available for all VMs and their guest systems (see Figure 2). The (copied) HSI for a VM or a guest system is referred to as a **virtual HSI**.

Differences between SU /390 and SU x86 with respect to the HSI

390 SU /390 are systems with a /390 architecture.
The BS2000 operating system uses the HSI of the SU /390 (**the real HSI**) directly here.

x86 SU x86 are systems with an x86-64 architecture.
The BS2000 operating system uses the nonprivileged, real HSI of the SU x86 here and the privileged HSI which is provided by the firmware.

For further information on the technical structure of the SE servers and on the system environment, see the manual "Operation and Administration" [19].

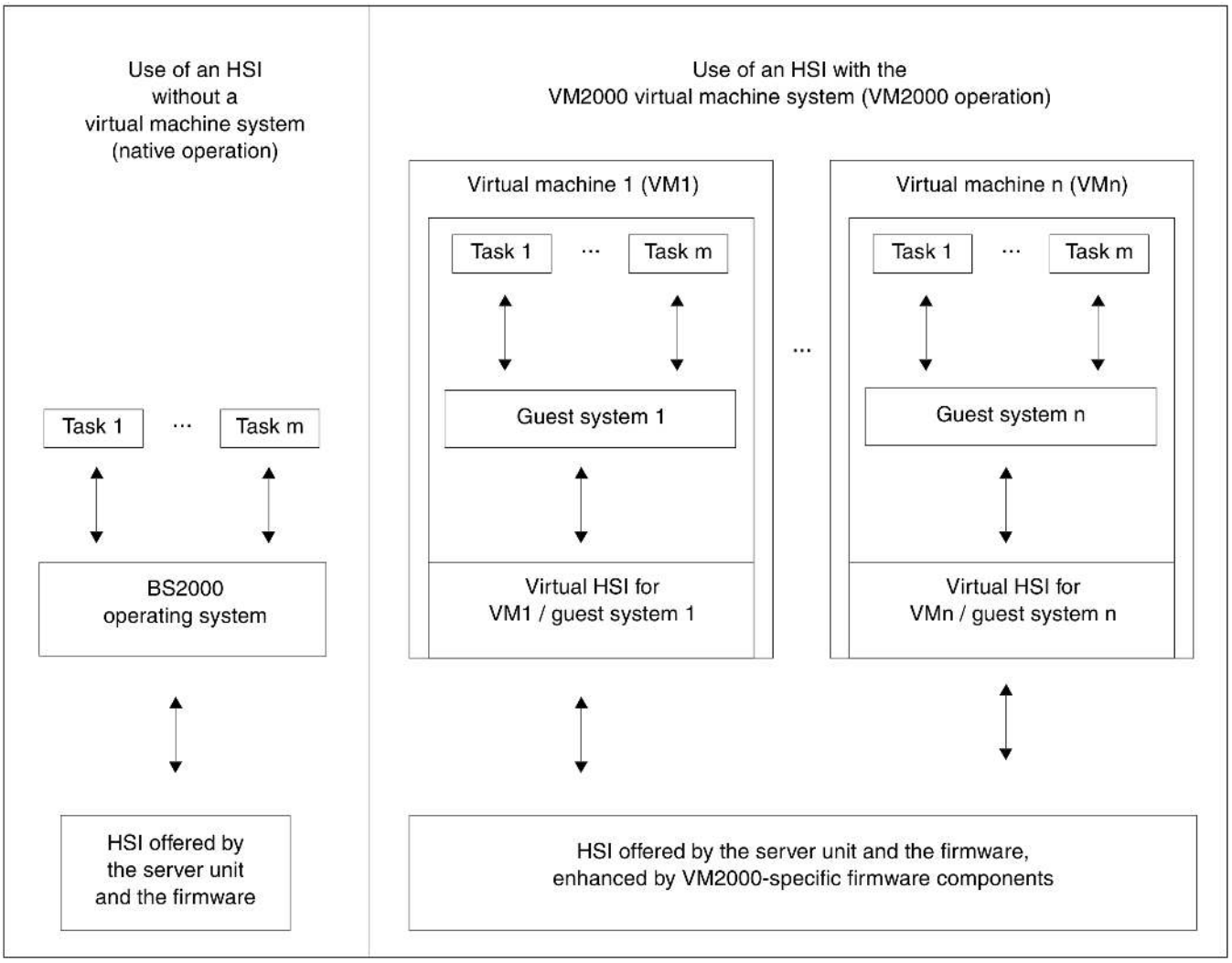


Figure 2: Use of the HSI of a Server Unit without and with VM2000

2.3 Implementation of VM2000

Depending on the architecture of the Server Unit, there are fundamentally different technical solutions for VM2000.

Implementation principle for SU /390

On SU /390, VM2000 controls the hardware of the Server Unit.

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.

Detailed information can be found in the section "Implementation on SU /390".

Implementation principle for SU x86

On SU x86, the X2000 carrier system controls the hardware of the Server Unit.

The VM2000 monitor manages the VMs with the guest system BS2000 (**BS2000 VM**) and provides its functions via the VM2000 user interface.

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 SE servers is the manual “Operation and Administration” [19]. A detailed description of working with the SE manager is also provided there.

Detailed information can be found in the section "Implementation on SU x86".

2.3.1 Implementation on SU /390

For VM2000 operation, VM2000 functions have been added to the HSI offered by the firmware of the Server Unit. The extended HSI is referred to as the **VM2000 HSI** (see Figure 3).

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).

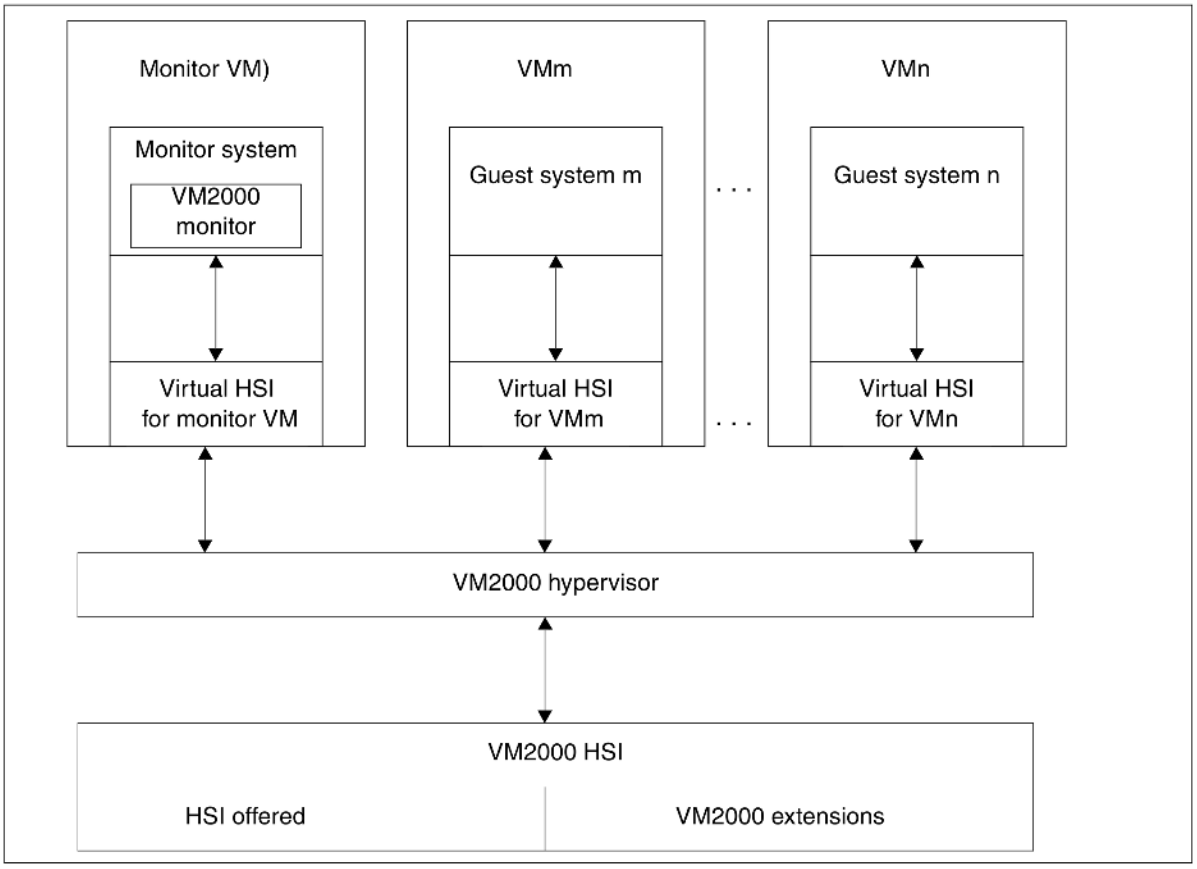


Figure 3: Structure of VM2000 on SU /390

The VM2000 hypervisor performs the following tasks:

- Allocate the CPUs to the individual VMs
- Handle global interrupts
- Control machine error recovery
- Emulate privileged instructions
- Coordinate I/O operations for all shared devices
- Process monitor jobs

In the VM2000 HSI, two processor states are possible (see Figure 4):

- 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 Server Unit's hardware registers each time the processor state is activated. This context consists of

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

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

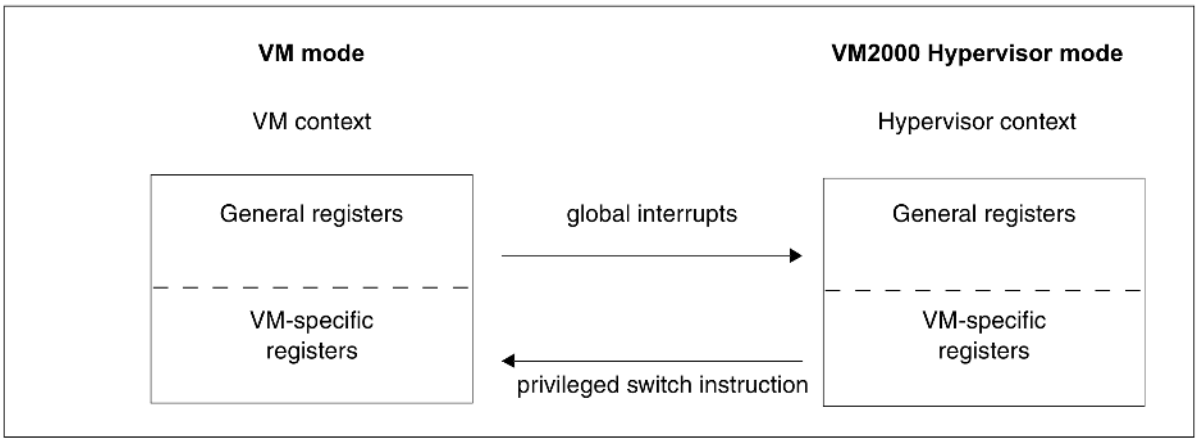


Figure 4: Processor states and context on SU /390

VM2000

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

Before VM2000 mode is initialized, the operating mode "VM2000 mode" must be set with the SE Manager, see "Operating modes". The monitor system can then be started, see "Starting the monitor system".

The VM2000 functions and commands offered are described in this manual. Fundamental VM2000 functions are also available to the SE server administrators via the SE Manager, see section "VM2000 functions in the SE Manager".

The full VM2000 functionality is available to the VM2000 and VM administrators via the user interface of VM2000 (see "VM2000 Operating strategy").

The BS2000 guest systems can work together with VM2000, see section "VM2000 in the guest systems".

2.3.2 Implementation on SU x86

Domains

On the Server Unit 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 I/O drivers in domain 0 operate and virtualize the Server Unit. The Xen hypervisor controls execution of all the domains (scheduling).

BS2000 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. It provides the operating functionality for devices (the BS2000 device emulation), and executes the physical inputs/outputs.

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

VM2000

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

Before VM2000 mode is initialized, the operating mode "VM2000 mode" must be set with the SE Manager, see "Operating modes". 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 monitor system can then be started, see "Starting the monitor system".

> ¡ 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.

The VM2000 functions and commands which are offered for BS2000 VMs are described in this manual. Fundamental VM2000 functions are also available to the SE server administrators via the SE Manager, see section "VM2000 functions in the SE Manager".

BS2000 guest systems can work together with VM2000, see section "VM2000 in the guest systems".

For hardware-related implementation of the functions, VM2000 utilizes the carrier system X2000. 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, each BS2000 guest system requires a firmware component which converts the /390 code to the relevant x86-64 code of the Server Unit (CISC-FW) and offers the privileged HSI for the BS2000 guest system (HAL).

Each BS2000 VM of the Server Unit has its own such firmware component.

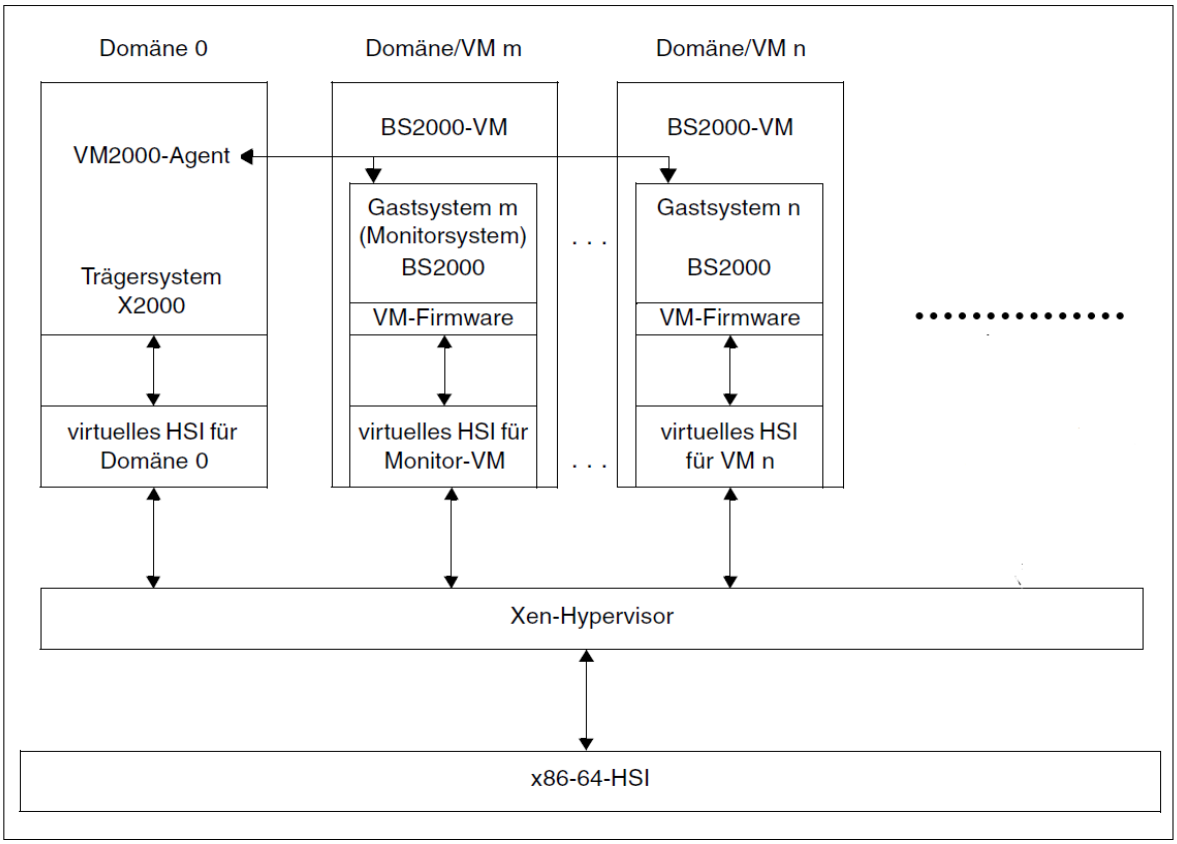


Figure 5: VM2000 on SU x86

Operating levels

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

- The full VM2000 functionality is available to the VM2000 and VM administrators via the user interface of VM2000 (see "VM2000 Operating strategy").
- Fundamental VM2000 functions are also available to the SE server administrators via the SE Manager, see section "VM2000 functions in the SE Manager".

Information on global resources in VM2000 commands

The VM2000 commands also supply information relating to resources which the BS2000 VMs share with the carrier system (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 pool of domain 0 (*POOL0).
- The output of `/SHOW-VM-RESOURCES INFORMATION=*MEMORY,ATTRIBUTE=*ALL/*ASSIGNED` also supplies the amount of main memory occupied by the domain 0 (`FOREIGN VMS`).
- `/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 (*POOL0).
- `/SHOW-VM-STATUS INFORMATION=*TIMER` also supplies information on the load on domain 0 in *POOL0.

Main memory

The Server Unit’s main memory contains a fixed portion for domain 0. The remaining main memory can be used for BS2000 VMs.

The planning for the assignment of the real main memory for VM2000 operation is the task of the Server Unit administration.

Information on the distribution 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 main memory of the Server Unit, but spread.

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 SU x86".

CPUs, CPU pools

The real CPUs of the Server Unit are subdivided into static CPU pools during system initialization of the Server Unit. 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 “Operation and Administration” [19] are specified in parentheses):

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

  The initial number of CPUs in the standard CPU pool is the same as the number of BS2000 CPUs of the Server Unit, including the extra CPUs (license required). VM2000 manages and reconfigures only the CPUs of the standard CPU pool, see "Real CPUs".

With VM2000 further (dynamic) CPU pools can be created. See section "Managing CPU pools".

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 connected to SU x86".

2.4 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) and of the VMs' virtual CPUs
- Management of the CPU pools
- Planning and controlling the distribution of CPU capacity to the VMs
- Assigning CPUs to a VM (scheduling, SU /390)

Reconfiguration of channels, control units and paths

- Reconfiguring Channels, controllers and paths (SU /390)

Management and reconfiguration of the storage media

Memory management of VM2000 includes:

- Managing main memory
- Reconfiguring main memory

Peripheral device management

The following are given particular consideration in peripheral device management:

- Assigning and removing devices of a VM
- Assigning and removing pubsets of a VM
- Special devices
- Shared disk devices
- Parallel access volume (PAV, SU /390)
- Using the Highspeed Net Connect HNC (SU /390)
- Dynamic I/O configuration modification

2.5 Supported hardware and operating systems

VM2000 V12.0 can be used on the Server Units /390 (SU /390) and x86 (SU x86) of the current SE servers.

> Further capacity levels of the SE 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 V12.0 operates all peripheral devices which are supported by BS2000 OSD/BC V11.0 and higher.

VM2000 V12.0 is executable in a monitor system with BS2000 V21.0.

The following guest system software can run under VM2000 V12.0:

- BS2000 OSD/BC V11.0 (software package OSD/XC V11.0)
- BS2000 V21.0
- Successor versions to BS2000 V21.0, provided they have been released for use under VM2000 V12.0 (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 media available for operating the guest systems under VM2000 (see "Operating levels of VM2000").

2.6 Constraints

- Constraints that apply to the hardware
- Constraints that apply to VM2000

2.6.1 Constraints that apply to the hardware

The Server Units of the SE servers exhibit certain differences from the full functional scope of VM2000 due to the hardware architecture, and these differences must be taken into consideration when using VM2000. These are indicated in the following list, and you are referred to relevant sections of the manual for more information.

The SU /390 exhibits the following differences:

- The minimum main memory size and the granularity of the memory allocation is 1 Mbyte (see "Managing main memory").
- The main memory must be at least 512 Mbytes in size for the monitor VM.

The SU x86 exhibits the following differences:

- Hot spare CPUs are not available.
  Cold spare CPUs can be configured, see "Implementation on SU x86".
- On SU x86, 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 "Managing main memory"). The main memory of a VM is no longer contiguous.
- The main memory must be at least 1024 Mbytes in size for a VM.

2.6.2 Constraints that apply to VM2000

When operating a Server Unit with VM2000, various constraints and limitations have to be taken into account that do not apply in native operation. These are indicated in the following list, and you are referred to relevant sections of the manual for more information.

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").

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 SU /390) the configuration file of the monitor VM (see "Setting the VM2000 parameters in the parameter service").
- A guest system cannot be started by an IPL device that is shared, i.e. in `SHARED` usage mode (see "Starting the guest system on a VM").
- The "Channel Subsystem Monitoring Facility" function (SU /390) for recording detailed device-specific operating times 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 "Measurement with the openSM2 measurement monitor").
- For shareable devices that are assigned to more than one VM (`SH(I)` usage mode), the Channel Subsystem Monitoring Facility (SU /390) is **not** activated. The openSM2 performance measurement monitor does not receive any measurement data for these devices (see "Measurement with the openSM2 measurement monitor").
- The BS2000 command `/WRITE-IOCF` (SU /390) can only be executed in the monitor system.

The following software prerequisites may need to be satisfied:

- A prerequisite for using VM2000 on an SU /390 is the software product REWAS in the monitor system.
- A prerequisite for working with job variables is the software product JV in the monitor system. See "Monitoring a VM with job variables (MONJV)" and "Reconfiguring main memory".
- If the 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 "Mirror pubsets".
- The software product SECOS is required for assigning privileges to a user task (which is not TSOS) in the monitor system, see "Privileges of the user task in the monitor system".
- The software product OMNIS is recommended when the guest system is operated via $VMCONS or logical consoles are used, see "Establishing a connection to $VMCONS" and "Operating the guest systems using logical consoles".

2.7 Comparison of VM2000 on SU /390 and SU x86

Topic	SU /390	SU x86
Hardware Architecture of the Server Unit	/390 architecture	x86-64 architecture
CPUs of the Server Unit Max. number of real CPUs [1]	16	72
Hot spare CPUs, managed by VM2000	For all models of the SU /390	Not available; cold spare CPUs can be configured
Extra CPUs, managed by VM2000	For specific models of the SU /390	For specific models of the SU x86
Max. multiprocessor level of a VM [2]	16	32
Peripherals Operated peripherals	Fibre Channel peripherals	Fibre Channel peripherals
Operated devices	Devices as of BS2000 OSD/BC V11.0	Devices as of X2000 V6.4 and BS2000 OSD/BC V11.0
Dynamic I/O configuration modification	via BS2000 commands in the monitor system	via the SE manager
Main memory Main memory used	Real main memory of the Server Unit	Real main memory of the Server Unit
Smallest main memory unit managed	1 Mbyte	2 Mbyte
Main memory size of the monitor VM	At least 512 Mbytes, default: 512 Mbytes	At least 1024 Mbytes, default: 1024 Mbytes
Monitor system Operating system	BS2000 V21.0	BS2000 V21.0
IPL of the monitor system	Via SE Manager or automatically when the Server Unit is started	Via SE Manager or automatically when the Server Unit is started
Manual restart of the monitor system in the event of an error	Restart via SVP console in the SE Manager	(Diagnostic) IPL via SE Manager

Topic	SU /390	SU x86
Guest systems Operating system	BS2000 OSD/BC V11.0, BS2000 V21.0 and later versions if released for VM2000	BS2000 OSD/BC V11.0, BS2000 V21.0 and later versions if released for VM2000
Maximum number of VMs/guest systems [3]	15	32
BS2000 consoles for operating the guest system	Via SE manager (KVP)	Via SE manager (KVP)
IPL device	Disk 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 [4] 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 SE manager)

1 The numbers in this line reflect the maximum hardware configuration of current Server Units. VM2000 can manage up to 64 real CPUs. The number of real CPUs for VM2000 operation is further restricted by the BS2000 model installed.

2 Defined by the firmware and the VM2000 implementation. Further restricted by the BS2000 model installed.

3 VM2000 can manage up to 99 VMs.

4 From the startup parameter file, only the VM2000 parameter `VERSION` is analyzed.

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” [14] manual.

VM2000 consists of the release units VM2000-MON (VM2000 monitor), VM2000-UTIL (VMDUMP) and VM2000-HPV (VM2000 hypervisor, only for SU /390).

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 (depending on the architecture of the Server Unit)	SYSLNK.VM2000-MON.<ver> SKMLNK.VM2000-MON.<ver>
RMS selectable unit [2] for VM2000 monitor	Contains the REPs 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 (template file)	SYSPAR.VM2000-MON.<ver>.SSP
Program file for VM2000 hypervisor	Program file for the VM2000 hypervisor (SU /390)	SYSPRG.VM2000-HPV.<ver>
RMS selectable unit VM2000 hypervisor	Contains the REPs required for the VM2000 hypervisor (SU /390)	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>

Component	Meaning	Standard file name
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>

1 <ver> stands for the version identifier, e.g. 120 for VM2000 V12.0

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 (SU /390), 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 "Installing the VM2000 message files".

3.1 Generating and configuring the device peripherals

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

- Device peripherals on SU /390
- BS2000 devices connected to SU x86

3.1.1 Device peripherals on SU /390

Hardware generation

When hardware generation takes place for the Server Unit using IOGEN, an I/O configuration file (IOCF) is created which defines the entire I/O 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` 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 KVP consoles and LAN devices

In addition to the hardware generation, the KVP consoles and LAN devices must be generated with the same mnemonic device names on the MU and HNC using the SE Manager, see the “Operation and Administration” manual [19].

Generating virtual consoles

Devices which are to be assigned to the guest systems as virtual consoles must be generated as consoles with the device type code 02/03 in the ATTACHED state when hardware generation takes place.

The virtual consoles are generated exclusively on a virtual channel type S (CHPID=FE).

> ¡ We recommend that no virtual consoles should be generated. The guest systems should be operated with KVP consoles or logical consoles, see section "Operating the guest systems using BS2000 consoles".

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 I/O 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 states 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 “Introduction to System Administration” [2].

Changes to the set of devices through a dynamic change to the I/O configuration (see "Dynamic I/O configuration modification") are automatically taken into account by VM2000.

Handling unknown device type codes

If a guest system (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.

Local and remote communication

For BS2000, ZASLAN and LOCLAN are (communication) devices used for the LAN connection to the physical network or for internal communication of the Server Unit. Each device must be assigned to the appropriate guest system. In addition, the communication paths must be configured in the BCAM of all guest systems.

LOCLAN

This port is emulated by the HNC and, under VM2000, enables a LAN connection to the other VMs that are also connected to LOCLAN, via TCP/IP, see also section "Using the Highspeed Net Connect HNC (SU /390)".

The LOCLAN port can also be emulated by another MU. However, this only establishes a local connection to M2000, with a gateway to the communication components 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 HNC 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. For this to function, a separate ZASLAN connection must be configured **for every VM** during the hardware configuration.

3.1.2 BS2000 devices connected to SU x86

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 SU x86 are always X2000 emulations of real devices, channel connections being emulated for BS2000.

BS2000 devices, i.e. the emulated devices, are configured exclusively via the SE Manager, see the “Operation and Administration” manual [19]. No hardware generation takes place in BS2000.

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
- LAN devices 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 states 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 “Introduction to System Administration” [2].

Changes to the set of devices through a dynamic change to the I/O configuration (see "Dynamic I/O configuration modification") are automatically taken into account by VM2000.

Handling unknown device type codes

If a guest system (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.

Local and remote communication

For BS2000, ZASLAN and LOCLAN are (communication) devices used for the LAN connection to the physical network or for internal communication of the Server Unit. Each device must be assigned to the appropriate guest system. In addition, the communication paths must be configured in the BCAM of all 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 local connection to X2000 with a gateway to 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 Server Unit 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 **for every VM** using the SE Manager.

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.
> 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, 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 (SU /390)
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 (SU x86)
VM2T	Task for processing requests of an MU or for a VM Migration on the target SU (SU /390)

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 "Preparing to use $VMCONS") 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 "Preparing to use $VMCONS").

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`).

390 The following parameter is only relevant on SU /390:

LM-TASK-PRIORITY = *HIGH / <integer 60..255>

Specification of the priority for the VM2000 system tasks that control and perform a VM Migration in the monitor systems of the affected Server Units (source and target SU). The highest possible priority is the default value (*HIGH, corresponds to the value 60).If the priority of the VM2000 system tasks for controlling and performing VM Migrations is reduced, i.e. the value for LM-TASK-PRIORITY is increased, other tasks in the monitor system have a higher probability of being allocated CPU capacity. This can be especially important for a monitor system with the MONO multiprocessor level.

The operand names and keywords must not be abbreviated. If no specification is made or if the specifications are incorrect, 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 different VM2000 versions on BS2000

The VM2000 parameter VERSION (in SYSPAR.BS2.<ver>) controls the VM2000 version to be selected. The default setting for the monitor system BS2000 V21.0 is VERSION=V12.0.

See section "Setting the VM2000 parameters in the parameter service".

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 a current session, the message file which belongs to the started version of subsystem VM2-MON is used. If you want to prevent the superfluos message files from being activated, you can remove them from the MIP parameter file using /MODIFY-MIP-PARAMETERS. If required, removed entries can be added using /MODIFY-MIP-PARAMETERS again.
- An entry for the version-specific SDF syntax file is contained in the SDF parameter file for the version of VM2-MON installed most recently. When the monitor system is started, only this SDF syntax file is activated. If it is not the desired version of the SDF syntax file of VM2-MON, you can correct the entry in the SDF parameter file using /MODIFY-SDF-PARAMETERS.

The parameter files need not be cleared for VM2000 operation normally, mainly not in case you want to start the version of VM2-MON installed most recently.

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>
MSGNOFL=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 (SU /390)
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 SU x86
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 all messages using an HTML application on our manual server (https://bs2manuals.ts.fujitsu.com) in the current version of BS2000 instead of in the previous manual "System Messages" and on the DVD “BS2000 SoftBooks”.

You can also use the MSGMAKER utility routine (see “utility routines” user guide [3] to retrieve all messages from the VM2000 message file.

The guaranteed messages of VM2000 are listed in the related command descriptions, see also section "Guaranteed messages".

3.4 Installing the VM2000 hypervisor files (SU /390)

x86 On SU x86 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 "Initializing and terminating VM2000"). 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 with their standard file names (`SYSPRG.VM2000-HPV.<ver>`, `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 “Introduction to System Administration” [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 by the Customer Service without interrupting operation.

3.5 Setting the VM2000 parameters in the parameter service

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

VERSION = V12.0 / V`mm.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.
This specification is only necessary if the required VM2000 version is not the standard VM2000 version of the monitor system used.
The default value for BS2000 V21.0 is `VERSION=V12.0`.

390 The following parameters are only relevant on SU /390:

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).

HYPERVISOR = 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).

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

Specifies the size of the hypervisor main memory in Mbytes (see "Managing main memory").

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. An additional 1 Mbyte per 2 Gbytes main memory size is added for the execution of VM Migrations.

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 periodically with message `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).

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 "Assignment sets, implicit device assignment and release". 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 “Introduction to System Administration” [2] for more details on the startup parameter service.

> BS2000 offers automatic selection of the parameter file by startup. For the monitor system, the suffix from the IOCF generation (SU /390) or the system name (SU x86) is appended to the default parameter file. This also applies for guest systems under VM2000. If the guest system on an SU x86 is started via an automatic restart or via the SE Manager, the system name is appended as a suffix. In all other cases (especially of guest systems on SU /390), the VM name is added as a suffix. You will find more information in the “System Installation” manual [1].

Example of a VM2000 parameter set (SU /390)

```
/BEGIN VM2000------------------------------------------------------------------  (1)
* VM2000 PARAMETERS
CONFIGURATION=$MYUSERID.CONF.VM2000--------------------------------------------  (2)
/EOF---------------------------------------------------------------------------  (3)
```

(1) Start of the VM2000 parameter set.

(2) A version-independent parameter file is used to configure the monitor VM. The defaults are used for the remaining VM2000 parameters.

(3) End of the VM2000 parameter set.

3.6 Configuration settings for the monitor VM

To operate VM2000, a few configuration settings must be made for the monitor VM which depend on the architecture of the Server Unit:

- Configuration via the VM2000 configuration file (SU /390)
- Configuration using the SE Manager (SU x86)

3.6.1 Configuration via the VM2000 configuration file (SU /390)

On SU /390, the attributes and resources for the monitor VM are defined in the VM2000 configuration file.

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 "CREATE-VM (Initialize a VM)")

  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 512 Mbytes (default value).
  - Minimum main memory size of the monitor VM
    MIN-MEMORY-SIZE = *STD (default value) is recommended. Minimum: 512 Mbytes. The DIMENSION operand cannot be specified.
  - Maximum main memory size of the monitor VM (MAX-MEMORY-SIZE)
    This operand is of no significance.
  - 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 I/O 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..16>)
  Default value: *MAXIMUM, i.e. the monitor VM is created with the maximum possible number of virtual CPUs. The maximum number of virtual CPUs corresponds to the number of real normal CPUs in the Server Unit, but may not be greater than 16 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.
  >
  > If VM Migration is to be used, the multiprocessor grade *BI is recommended and expected for a performant process.

- Location of monitor the 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 "Initializing a VM".
  Assignment sets (ASSIGN-BY-GUEST=*YES) cannot be specified.
- Specification whether, in the case of fixed CPU assignment (dedicated CPUs), the monitor VM still retains control over a real CPU if its virtual CPU becomes inactive (interruptible wait state) (VM-ACTIVE-IDLE=*NO/*AT-DEDICATED-CPUS)
  Default value: *NO (the virtual CPU is removed).
- MONJV to monitor the VM (MONJV)
  The operand cannot be specified for the monitor VM.
- PERSISTENT attribute for persistent creation of the VM
  Only PERSISTENT=*NO may be specified for the monitor VM.

- /ADD-VM-DEVICES
  UNITS=list-poss(28):<alphanum-name 2..2>/<x-text 4..4>
  (see command description in section "ADD-VM-DEVICES (Allocate devices to a VM)")

  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.

Commands in the configuration file which do not refer to the monitor VM are ignored.
If `/CREATE-VM` 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

```
/CREATE-VM MEM-S=512,MIN-MEM-S=512,VM-N=MONITOR,CPU-Q=20,PROC=BI,PASS=    (1)
/REMARK HOME PUBSET (SPECIFICATION OPTIONAL) ———————————————————————    (2)
/ADD-VM-DEVICES VM-ID=1,UNITS=(9900,9901,9902) ———————————————————————    (3)
/REMARK REDUNDANT MAIN CONSOLES
/ADD-VM-DEVICES VM-ID=1,UNITS=(C2,C3,C4,C5)
/REMARK EMULATED LAN DEVICES OF REDUNDANT MANAGEMENT UNITS
/ADD-VM-DEVICES VM-ID=1,UNITS=(CC80,CC81,CD80,CD81)
/REMARK REDUNDANT HNC DEVICES
/ADD-VM-DEVICES VM-ID=1,UNITS=(CC40,CC41,CD40,CD41)
/REMARK SHARED DISK
/ADD-VM-DEVICES VM-ID=1,UNITS=990F,TYPE=*SD ———————————————————————    (4)
```

(1) Set up monitor VM.

(2) Comment line.

(3) Certain devices are assigned exclusively to the monitor VM.

(4) A device is assigned to be shared between the monitor VM and other VMs.

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.

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 above), and the KVP console and the devices of the home pubset are assigned to the monitor VM explicitly at startup.

3.6.2 Configuration using the SE Manager (SU x86)

The configuration settings for the monitor VM are made on SU x86 using the SE Manager, see the “Operation and Administration” manual [19].
Any existing configuration file for the monitor VM on the home pubset is ignored.

Setting the attributes and resources for the monitor VM

These settings must be specified via the SE 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 operation mode* tab of the *Systems > name of the SU x86* menu:

- Multiprocessor level (number of virtual CPUs, default value: 1)
- Current, minimum and maximum main memory size in Mbyte (default value: 1024)
- 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 SE 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 SE 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 SE Manager can be used to specify settings for automatic IPL of the monitor system and the remaining runtime for BS2000 shutdown.

The following settings for automatic IPL of the monitor system can be specified in the *VM options* tab of the *Systems > name of the SU x86* menu:

- IPL option (automatic IPL selected / not selected)
- IPL device (boot disk)
- IPL console (console device)
- Startup mode
- System name

> The IPL option is always reset (automatic IPL not selected) 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 in the *VM options* tab (default: 30 minutes). This time applies to the monitor system and all BS2000 guest systems when the Server Unit is shut down (see "Terminating the monitor system and VM2000").

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"), 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>),
   ```

   The user ID must be specified as the parameter VMCONS-OPERID in the parameter file of the VM2-MON subsystem (default value: VMADMIN), see "Setting up the VM2000 monitor".
   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 [11]) /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,
                            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 the guest systems, provided that these are released for use under VM2000 V12.0 (see Release Notices).

4 Initializing and terminating VM2000

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

- Initializing VM2000 operation
  - Operating modes
  - Starting the monitor system
- Starting up the monitor system
  - Required components
  - Startup procedure in the monitor system
  - Example of startup in the monitor system
- Terminating the monitor system and VM2000
  - Coordinated shutdown of VM2000
  - Shutdown in the monitor system
- Loading, operating and terminating guest systems

4.1 Initializing VM2000 operation

To initialize VM2000 operation, first the operating mode for VM2000 operation is set and then the monitor system is started.

- Operating modes
- Starting the monitor system

4.1.1 Operating modes

All Server Units have the operating modes “Native operation” and “VM2000 operation”.

The operating mode is set using the SE Manager in the main menu (Server Unit selected) in the *BS2000 operation mode* tab.

390 On the SU /390, the operating mode can also be set using the functions of the SVP console. The SVP console can be opened in the *SVP console* tab in the SE Manager (SU /390 selected).

When the Server Units are started, the operating mode of the previous session is set again.

Before switching the operating mode from “VM2000 operation” to “Native BS2000”, first all BS2000 guests systems and then the monitor system must be terminated, see section "Terminating the monitor system and VM2000".

Special considerations on the SU x86

On the SU x86, the operating mode can also be switched during ongoing operation. However, BS2000 systems must be terminated first.

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

Switching the operating mode to “VM2000 operation” terminates the domain `BS2000` and a domain with the domain name `MONITOR` and the startup settings for the monitor VM is configured, see "Configuration using the SE Manager (SU x86)".

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

4.1.2 Starting the monitor system

See also the “Operation and Administration” manual [19].

Starting the monitor system on an SU /390

The monitor system on an SU /390 can be started in various ways:

- When the operating mode is switched to "VM2000 operation" using the SE Manager (see "Operating modes"), the monitor system is started immediately.
- After the SU /390 has been started in the operating mode "VM2000 operation", the monitor system can be started as follows:
    - Automatically (after a corresponding setting has been made in the SVP)
    - Manually using the SE Manager in the main menu (monitor VM selected) in the *Operation* tab with the action *BS2000 IPL*
    - Manually using the functions of the SVP console. The SVP console can be opened in the SE Manager (SU /390 selected) in the *SVP console* tab.

Starting the monitor system on an SU x86

After the operating mode “VM2000 operation” has been set, the monitor system on an SU x86 can be started in different ways:

- Automatically when initializing the domain `MONITOR` (see "Configuration using the SE Manager (SU x86)")
- Manually using the SE Manager in the main menu (monitor VM selected) in the *Operation* tab with the action *BS2000 IPL*
- Manually using SVP functions of the KVP console which is assigned to the monitor VM

4.2 Starting up the monitor system

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

- Required components
- Startup procedure in the monitor system
- Example of startup in the monitor system

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	Like BS2000
Startup for BS2000	System initialization for BS2000 and VM2000 (monitor system)			x
Organizational program for BS2000	Organizational program of the monitor system, Standard-EXEC			x
Startup parameter file	Contains parameters for BS2000 and VM2000 operation (see "Setting the VM2000 parameters in the parameter service")		x	
REP loader for BS2000	Contains system corrections for BS2000			x
390 Hypervisor file for VM2000	Program file for the VM2000 hypervisor (see "Installing the VM2000 hypervisor files (SU /390)") on SU /390	x		
390 Hypervisor REP file	Contains system corrections for the VM2000 hypervisor on SU /390	x		
390 Configuration file for the monitor VM	Required on SU /390 if the default settings and the default device configuration of the monitor VM are to be changed (see "Configuration via the VM2000 configuration file (SU /390)")	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, apart from some additional actions for VM2000.
The “Introduction to System Administration” [2] provides a more detailed description.

390 On SU /390, the startup of the monitor system is terminated if initialization of VM2000 is faulty.

x86 On SU x86, the 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 "Setting the VM2000 parameters in the parameter service").

> i 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 (SU /390).

The attributes and resources of the monitor VM were set when VM2000 was installed, see chapter "Installing VM2000".

390 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 V12.0B READY ***

NSI0000 *** S Y S T E M   R E A D Y ***
```

390 If VM2000 on SU /390 can work with VM definitions, after `BCAM ACTIVE` the following message is displayed:

```
VMS2330 STORAGE FOR VM DEFINITIONS ACCESSIBLE
```

Dialog startup

In the case of a dialog startup, after the corrections for the operating system have been requested, entries are requested on SU /390 for the VM2000 hypervisor and the monitor VM.

x86 On SU x86, the attributes and resources for the monitor VM are set via the SE manager, see section "Configuration using the SE Manager (SU x86)".
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 the entries for VM2000, 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 with the standard file name `SYSPRG.VM2000-HPV.<ver>` is loaded.
P.<filename>	A VM2000 hypervisor file which is created as a PAM file with the name `<filename>` is loaded (see "Installing the VM2000 hypervisor files (SU /390)").

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

```
NSI0050 SPECIFY VM2000 HYPERVISOR REP 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 with the standard file name `SYSREP.VM2000-HPV.<ver>` is processed (see "Installing the VM2000 hypervisor files (SU /390)").
P. FN=<filename>	A VM2000 hypervisor REP file created as a SAM file with the file name `<filename>` is processed.
P.CONS	Direct entry of REPs at the BS2000 console. The dialog is the same as when entering REPs in native operation.
P.END	Entry of the REPs is terminated. Startup continues.

> More than one file can be processed. Additional REPs 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 with the standard file name `SYSPAR.VM2000-MON.<ver>` is processed.
P.FN=<filename>	The configuration file `<filename>` is processed.
P.CONS	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` and `/ADD-VM-DEVICES` can then be specified for the monitor VM ("Configuration via the VM2000 configuration file (SU /390)"). 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 "Configuration via the VM2000 configuration file (SU /390)").
- 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 V21.0B on an SU /390. The output shown omits the parts which do not affect VM2000 (indicated by "...").

```
%P.NSI00E3 IPL-REPS READ: 0; EXECUTED: 0
%P.NSI1100 IPL DEVICE = 4IVV.0; IPL PATH = 9900 (MN=9900)
%P.NSI1106 *** BS2000 DIALOGUE STARTUP ***
?P.NSI1110 ENTER OPTIONS OR EOT. REPLY (UNLOCK,TEST,ALLDISK,DRV-SELECT,
           CREATE-DRV,IPL-CONF)
p.
%P.NSI3135 IPL DISK-SETUP READ FROM IPL-CONF PREPARED <date> <time>
%P.NSI1143 003 PUBVOLS OF HOME-PVS 4IVV WITH SYSID 174 ONLINE
%P.NSI1145 4IVV.0 <date> <time> ON 9900
%P.NSI1145 4IVV.1 <date> <time> ON 9901
%P.NSI1145 4IVV.2 <date> <time> ON 9902
%P.NSI1153 STATE OF PROCESSORS ONLINE:
%P.NSI1155 CPU 00 ONLINE, ATTACHED (IPL CPU)
%P.NSI1158 CPU 01 ONLINE, DETACHED
%P.NSI1158 CPU 02 ONLINE, DETACHED
%P.NSI1158 CPU 03 ONLINE, DETACHED
%P.NSI1158 CPU 04 ONLINE, DETACHED
%P.NSI1163 LOCAL DATE = <date>, TIME = <time> FROM SVP
%P.NSI1180 LOAD ADDRESS OF SYSPRG.STRT.210 = 0077E000
%P.NSI00E6 SYSREP.STRT.210 IS USED AS REP FILENAME
?P.NSI0050 SPECIFY PARAM FILE OR DEVICE.
           REPLY (EOT (USE STANDARD FILE);FN=FILENAME(,VOL=VSN);CONS;END)
p.
%P.NSI00E6 SYSPAR.BS2.210.SU710001 IS USED AS PARAM FILENAME
%P.NSI008A PROCESSING PARAMETER ADD FILE PARAMS.VM2000 ——————————————————  (1)
?P.NSI0050 SPECIFY PARAM FILE OR DEVICE.
           REPLY (EOT (USE STANDARD FILE);FN=FILENAME(,VOL=VSN);CONS;END)
p.end
?P.NSI1190 ENTER BS2000-FILENAME.
           REPLY (FILENAME(,VOL=VSN); EOT (USE STANDARD FILE))
p.
?P.NSI0050 SPECIFY BS2000 REP FILE OR DEVICE.
           REPLY (EOT (USE STANDARD FILE);FN=FILENAME(,VOL=VSN);CONS;END)
p.
%P.NSI00E6 SYSREP.BS2.210 IS USED AS REP FILENAME
?P.NSI0050 SPECIFY BS2000 REP FILE OR DEVICE.
           REPLY (EOT (END); FN=FILENAME(,VOL=VSN);CONS;END)
p.end ————————————————————————————————————————————————————————————————  (2)
%P.NSI0028 AVAILABLE MEMORY = 15.5 GB. CPU = 30029001 36000000
%P.NSI5103 BS2000 : NAME = <name>, VERSION = 21.0B00,
           GEN-TIME = <date> <time>
```

```
?P.VMS5005 ENTER VM2000 HYPERVISOR FILE NAME.
           REPLY (<FILE NAME>; EOT(USE STANDARD FILE))
p. ──────────────────────────────────────────────────────────────────────── (3)
%P.VMS5015 VM2000 HYPERVISOR LOADED FROM FILE 'SYSPRG.VM2000-HPV.120'
?P.NSI0050 SPECIFY VM2000 HYPERVISORREP FILE OR DEVICE.
           REPLY (EOT (USE STANDARD FILE);FN=FILENAME(,VOL=VSN);CONS;END)
p. ──────────────────────────────────────────────────────────────────────── (4)
%P.NSI00E6 SYSREP.VM2000-HPV.120 IS USED AS REP FILENAME
?P.NSI0050 SPECIFY VM2000 HYPERVISORREP FILE OR DEVICE.
           REPLY (EOT (END); FN=FILENAME(,VOL=VSN);CONS;END)
p.end ────────────────────────────────────────────────────────────────────── (5)
?P.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)
%P.VMS5014 FILE 'SYSPAR.VM2000-MON.120' IS USED AS
           VM2000 CONFIGURATION FILE ──────────────────────────────────── (8)
?P.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)
?P.VMS5021 ISSUE VM2000 COMMAND (MAX. 70 CHARACTERS) OR 'END'.
           REPLY (DESIRED INPUT; EOT=END) ──────────────────────────── (10)
p.add-vm-devices units=(fc10,fc11,fc12),vm-id=1 ──────────────────────── (11)
?P.VMS5021 ISSUE VM2000 COMMAND (MAX. 70 CHARACTERS) OR 'END'.
           REPLY (DESIRED INPUT; EOT=END)
p. ──────────────────────────────────────────────────────────────────────── (12)
?P.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)
%P.VMS5049 NUMBER OF REAL CPU(S) RUNNING:  4 ──────────────────────────── (14)
%P.VMS5050 NUMBER OF AVAILABLE SPARE-CPU(S):  1
%P.VMS5020 STARTUP OF THE MONITOR SYSTEM IN BIPROCESSOR MODE ──────────── (15)
%P.VMS5016 TOTAL MAIN MEMORY FOR VM2000:   15872 MB;
           MONITOR VM SIZE:   512 MB; HYPERVISOR SIZE:  16 MB ────────── (16)
%P.EMM2960 EFFECTIVE SYSSIZE VALUE IS 240 MB
%P.NSI5150 CPU'S ONLINE: 2  CPU'S INSTALLED: 4
...
%P.NSI3130 IPL-CONF SAVED ONTO DISK MN=9900 WITHOUT ERRORS
%E.EMM2301 PAGING AREA ON VOLUME 4IVV.0 SIZE = 00875.00 MB.
%E.EMM2301 PAGING AREA ON VOLUME 4IVV.1 SIZE = 00875.00 MB.
%E.EMM2301 PAGING AREA ON VOLUME 4IVV.2 SIZE = 00875.00 MB.
%E.EMM2850 THE SIZE OF THE PAGING-AREA IS 0002625.00 MB;
           THE RESERVED SIZE OF THE PAGING-AREA IS 0000011.60 MB.
%P.NSI6102 BS2000 EXECUTIVE: CLASS2 ACTIVE
%E.EMM2310 THE SIZE OF MAIN MEMORY IS 00000512 MB;
```

```
           THE MINIMAL SIZE OF MAIN MEMORY IS 00000512 MB
...
NMH1102 MESSAGE OUTPUT FILE ':4IVV:$TSOS.SYSMES.VM2000-UTIL.120',
        ACCESS=ISAM, ACTION=ADD ──────────────────────────────────────── (17)
NMH1102 MESSAGE OUTPUT FILE ':4IVV:$TSOS.SYSMES.VM2000-MON.120',
        ACCESS=ISAM, ACTION=ADD
...
BLS0990 PROCESSING OF REP FILE ':4IVV:$TSOS.SYSREP.VM2000-MON.120' STARTED
...
HLV0001 VMHEL FILE ':4IVV:$TSOS.SYS.VMHEL.<date> <time>' OPENED
        WITH 'SPACE=  402' ──────────────────────────────────────────── (18)
VMS5541 *** VIRTUAL MACHINE SYSTEM VM2000 V12.0B READY *** ──────────── (19)
ESM0501 'VM2-MON /V12.0' COPYRIGHT (C)
        '2023',FUJITSU TECHNOLOGY SOLUTIONS GMBH. ALL RIGHTS RESERVED
ESM0400 'CREATE' OR 'RESUME' SUBSYSTEM 'VM2-MON /V12.0' WITH
        'SYNCHRONOUS=*YES' AND 'RESET=*NO' ──────────────────────────── (20)
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)
...
/dcs dcsof=sof.su390 ──────────────────────────────────────────────────── (24)
...
VMS1030 BCAM APPLICATION '$VMCONS' READY ──────────────────────────────── (25)
...
VMS2330 STORAGE FOR VM DEFINITIONS ACCESSIBLE ─────────────────────────── (26)
...
/show-vm-resources information=*configuration ──────────────────────────── (27)
MAX NUMBER OF VMS                     15
MAX NUMBER OF VM-GROUPS                8
MAX NUMBER OF CPU-POOLS                4
MAX NUMBER OF CPUS PER VM              4
TOTAL REAL MEMORY SIZE (MB)        15872
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                     NONE
SPARE                     04(DET)
VMS3000 VM2000 COMMAND /SHOW-VM-RESOURCES FOR VIRTUAL MACHINE ( ,*VM2000)
        PROCESSED SUCCESSFULLY
NBR0740 COMMAND COMPLETED 'SHOW-VM-RESOURCES';
        (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
/show-vm-resources information=*all ──────────────────────────────────── (28)
             CPU-POOL(P)
```

```
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 1 MONITOR   *STDPOOL(P)  20.00  100.00  2(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 1 MONITOR   RUNNING         V21.0B  NO  NO    AS
F R E E      VM: 14          LARGEST-DOMAIN(MB):   15344
VM-ID        CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
 1 MONITOR   *STDPOOL            20.00/ 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)
CPU-ATTR     CPU-POOL     REAL CPUS
NORMAL       *STDPOOL     00(ATT),01(ATT),02(ATT),03(ATT)
EXTRA                     NONE
SPARE                     04(DET)
VM-ID        VM-MONJV
             NO VM-MONJV ASSIGNED
VM-ID        UNITS
 1 MONITOR   ASSIGNED BY VM-ADMIN
             EXCL:   C2,C3,C4,C5,9900,9901,9902,CC40,CC41,CC80,
                     CC81,CD40,CD41,CD80,CD81,FC10,FC11,FC12———————— (29)
             SHARED: 990F
DISPOSABLE   UNASS:  CA(VC),CB(VC),CC(VC),CH,CI,CJ,CK,CL,CM,CN,CO,
                     CS,CT,C0,C1,DH,DI,DJ,DK,DL,DM,DN,DO,DS,DT,N3,
                     N4,N8,TA,TB,TF,T0,T1,T2,9903,9904,9905,9906,
                     9907,9908,9909,990A,990B,990C,990D,990E,9910,
...
                     FC19,FC1A,FC1B,FC1C,FC1D,FC1E,FC1F
             SHARED: 990F
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR   00000000         16
 1 MONITOR   00000010        512        512
NONE         00000210      15344
LARGEST FREE DOMAIN:       15344
VM-ID        PRIV    DETAILS
 1 MONITOR   AS
VMS3000 VM2000 COMMAND /SHOW-VM-RESOURCES FOR VIRTUAL MACHINE ( ,*VM2000)
        PROCESSED SUCCESSFULLY
NBR0740 COMMAND COMPLETED 'SHOW-VM-RESOURCES';
        (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
```

---

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

(2) The REP entry for BS2000 is terminated.

(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 specifications of the configuration file are logged (only) in the CONSLOG file.

(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` are permitted.

(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.

(17) The message files of VM2000 are enabled.

(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.

(24) The data communication system is started in the monitor system.

(25) $VMCONS is started automatically.

(26) The system disk for VM definitions is available.

(27) Command of the VM2000 administrator for outputting the VM2000 configuration information, see "SHOW-VM-RESOURCES (Output status of VM2000 resources)".

(28) Command of the VM2000 administrator for outputting the status information on all VM2000 resources, see "SHOW-VM-RESOURCES (Output status of VM2000 resources)".

(29) In addition to the devices from the configuration file, VM1 is automatically explicitly assigned the devices of the home pubset when it is started, see "Configuration via the VM2000 configuration file (SU /390)".

4.3 Terminating the monitor system and VM2000

Because of the technical implementation of VM2000 (see "Implementation of VM2000"), termination of the monitor system has different effects on VM2000 and the Server Unit:

390 On SU /390, the shutdown of the monitor system also terminates operation of VM2000 and the Server Unit.

x86 On SU x86, VM2000 operation is terminated when the monitor system shuts down. The firmware component of the monitor system continues to run after the latter has shut down until the Server Unit has been shut down.
The Server Unit also continues to operate after the monitor system has shut down; it can be shut down using the SE Manager.

A new IPL can then take place on all Server Units.

Termination on SU /390

VM2000 and Server Unit operation are shut down on SU /390 in different ways:

- Using the VM2000 command `/SHUTDOWN-VM VM-ID=*VM2000(...)`.
  This is the recommended way to shut down VM2000, see "Coordinated shutdown of VM2000" (coordinated shutdown of all guest systems and of the monitor system without restart).
- Using the BS2000 command `/SHUTDOWN` in the monitor system (see "Shutdown in the monitor system") or by means of the action *BS2000 Shutdown* for the monitor VM in the SE Manager. 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 SU x86

There are two reasons for regular termination of VM2000 operation on SU x86:

- The Server Unit is to be shut down.
  Shutdown of the server unit is initiated via the SE manager, see manual “Operation and Administration” [19].
  Time-driven startup and shutdown of the Server Unit can also be configured via the SE 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 "Coordinated shutdown of VM2000".

  Finally carrier system operation is terminated and the Server Unit is switched off.
- The operating mode “Native operation” is to be switched to (see "Operating modes").
  In this case VM2000 operation must first be terminated (preferably by means of a coordinated shutdown of VM2000, see "Coordinated shutdown of VM2000"). Only then can the operating mode be changed.

VM2000 operation is also terminated using the BS2000 command `/SHUTDOWN` in the monitor system (see "Shutdown in the monitor system") or by the action *BS2000 Shutdown* for the monitor VM in the SE 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. 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. All VMs are also terminated. A detailed description of coordinated shutdown is contained in the section "Shutdown in the guest system and of VM2000".

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 initialized, irrespective of the current system state (`CHECK-VM-STATE=*NO`)
- Accounting for the monitor VM
- Detaching of the attached extra CPUs

390 On SU /390, all persistent VM definitions are updated by VM2000. The non-persistent VM definitions are deleted. These also include the VM definition of the monitor VM. For this action BCAM must still be active in the monitor system. Consequently, when the combined shutdown of VM2000 takes place, the `ACTION=*SHUTDOWN(QUIET-TIME=0)` operand should be selected.

x86 VM definitions are administered by X2000 on SU x86. No action by VM2000 when shutdown takes place.

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)        512   (100)
 2 SYSTEM2   *STDPOOL(P)   30.00  100.00  2(+1)        512   (100)
 3 SYSTEM3   *STDPOOL(P)   30.00  100.00  2(+1)        512   (100)
VM-ID        STATE            VERSION PER ADMIN PRIV
 1 MONITOR   RUNNING          V21.0B  NO  YES   AS
 2 SYSTEM2   RUNNING          V20.0B  NO  NO    AS
 3 SYSTEM3   RUNNING          V20.0B  NO  NO    AS
F R E E      VM: 12           LARGEST-DOMAIN(MB):   14320
```

```
/shutdown-vm vm-id=*vm2000(waiting-time=*unlimited),
                           action=*shutdown(quiet-time=0) ————————————  (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)
*** ...
*** EXC0557 SHUTDOWN PROCESSING COMPLETED
***
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (3,SYSTEM3),
           CODE 80FFFFFF, CPU 01
%  VMS2033 'EXC0557 SHUTDOWN PROCESSING COMPLETED' FROM VM (3,SYSTEM3)
           VIA SVP
%  VMS2051 GUEST SYSTEM ON VM (3,SYSTEM3) DOWN; REASON: SHUTDOWN
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (3,SYSTEM3),
           CODE 80FFFFFF, CPU 00
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,SYSTEM2),
           CODE 80FFFFFF, CPU 01
%  VMS2033 'EXC0557 SHUTDOWN PROCESSING COMPLETED' FROM VM (2,SYSTEM2)
           VIA SVP
%  VMS2051 GUEST SYSTEM ON VM (2,SYSTEM2) DOWN; REASON: SHUTDOWN ————————  (4)
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,SYSTEM2),
           CODE 80FFFFFF, CPU 00
***
*** Guest system operating in the monitor system:
*** ...
*** NRT1201 SHUTDOWN REQUESTED BY 'VM2000' ——————————————————————————  (5)
*** ...
*** EXC0557 SHUTDOWN PROCESSING COMPLETED
*** VMS0000 MONITOR SYSTEM TERMINATED. VM2000 TERMINATION
***
NRTT201 TASK TERMINATION DUE TO /SHUTDOWN  COMMAND ————————————————————  (6)
EXC0419 /LOGOFF AT 1137 ON <date> FOR TSN '0GLI'
```

---

(1) The command shows the initial situation for this example:
In addition to the monitor system, two further VMs with a BS2000 guest system execute. 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.

(3) The shutdown (`MODE=*END`) is initiated in the guest system. The message `NRT1201` is output by the shutdown processing in the guest system.

(4) Guest systems SYSTEM2 and SYSTEM3 have been shut down. The VM2000 administrator is informed of this. Asynchronous messages are marked with %.

(5) After all guest systems have been terminated, the shutdown is initiated in the monitor system (MODE=*END). The message NRT1201 is output by the shutdown processing in the monitor system.

(6) 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 “Introduction to System Administration” [2]) or by the action *BS2000 Shutdown* for the monitor system in the SE manager.

Before the monitor system is terminated, the operator should terminate each guest system by means of the BS2000 command /SHUTDOWN. The operator of 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 initialized, irrespective of the current system state (CHECK-VM-STATE=*NO)
- Accounting for the monitor VM
- Detaching of the attached extra CPUs

390 On SU /390, all persistent VM definitions are updated by VM2000. The non-persistent VM definitions are deleted. These also include the VM definition of the monitor VM. For this action BCAM must still be active in the monitor system. On SU /390 use of the /SHUTDOWN MODE=*QUIET or /BCEND command should therefore be avoided in the monitor system when shutting down VM2000.

x86 VM definitions are administered by X2000 on SU x86. No action by VM2000 when shutdown takes place.

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 SU /390) 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”) is output.

i In the case of /SHUTDOWN with restart, the monitor system must be prepared for automatic restart (see the “Introduction to System Administration” [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 by the action *BS2000 Shutdown* in the SE 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` or in the SE Manager (VM selected) in the *Operation* tab by the action *BS2000 IPL* (see "Starting the guest system on a VM").

  x86 On SU x86, the guest system can also be loaded using the SVP functions of the KVP console which is assigned to the VM (see the manual “Operation and Administration” [19]).

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 "Operating the guest systems using BS2000 consoles").

  When BCAM is 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 [12]).

  A guest system can also be operated using $VMCONS (see "Operating the guest systems (VC dialog via $VMCONS)").

  If the dialog system is ready for operation, operator commands can also be executed by user tasks with the OPERATING privilege.
- See "Monitoring under VM2000" 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 "SHUTDOWN-VM (Execute a shutdown in the guest system and for VM2000)"). As in native operation, it can also be shut down using the BS2000 command `/SHUTDOWN` (see the “Introduction to System Administration” [2]).

  BS2000 shutdown for a guest system can also be initiated using the SE Manager, VM selected, *Operation* tab, *BS2000 Shutdown* action.
- 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 and the operating functions of VM2000:

- VM2000 Operating strategy
- Operating a VM
- VM Migration
- Working with VM definitions
- Operating VM groups (SU /390)
- Operating the whole VM2000 system
- VM2000 in the guest systems

The operating functions are realized by 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".

User interfaces

There are two user interfaces for operating VM2000 and the virtual machine:

- The full VM2000 functionality is available to the VM2000 and VM administrators via the user interface of VM2000 (which is described in this chapter).
- Fundamental VM2000 functions are also available to the SE server administrators via the SE Manager, see section "VM2000 functions in the SE Manager".

5.1 VM2000 Operating strategy

The VM2000 operating strategy describes

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

5.1.1 Operating levels of VM2000

VM2000 has three operating levels. Figure 6 provides an overview of these.

Operating level 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 "Types of command"). The VM2000 administrator 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.

The whole VM2000 system can also be operated from a privileged user task in the monitor system or (e.g. when BCAM is not active in the monitor system) via BS2000 consoles in the monitor system (see "Operating the guest systems using BS2000 consoles").

Operating level Virtual machine (VM administration)

The VM administrator operates a VM. 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 "Privileges of the user task in the monitor system").

The ADMIN dialog can also be conducted by means of the $VMCONS administration application that runs in the monitor system (see "Operating the virtual machines (ADMIN dialog via $VMCONS)"). The connection to $VMCONS is established by means of an appropriate DCAM application. OMNIS will usually be used for access from 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 "Operating the guest systems using BS2000 consoles"). See also the section entitled “Functions of the operator” in the “Introduction to System Administration” [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.

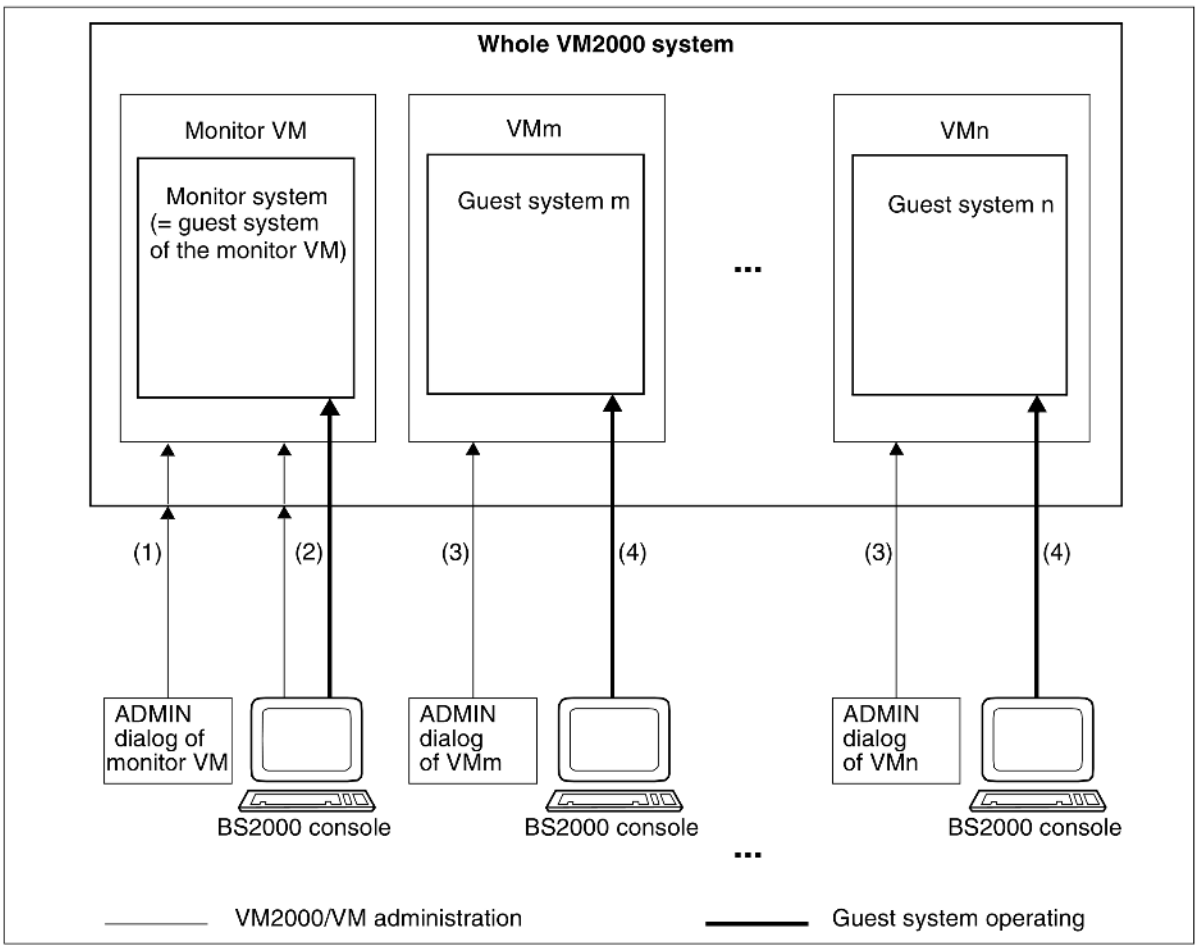


Figure 6: Operating strategy of VM2000 (recommendation)

(1) All VM2000 commands can be entered via the ADMIN dialog of the monitor VM. The ADMIN dialog is carried out 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 carried out 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 carried our via privileged user tasks in the monitor system or via the administration application $VMCONS which runs in 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 [11]).

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".

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".

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 with this TSN in the CONSLOG file of the monitor system.
- Enter VM2000 commands to operate the VM (see "Operating a VM" and "VM2000 commands") or the VM2000 system as a whole (see "Operating the whole VM2000 system" and "VM2000 commands"). 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.

> For **one** user task, only **one** ADMIN dialog can be carried out 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 “Introduction to System Administration” [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   VMS2320   VMS2411 - VMS2422   VMS2424   VMS2425
VMS2430 - VMS2434   VMS2440   VMS2441   VMS2450 - VMS2452   VMS4910   VMS4912
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   VMS6018   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`.

> **i** Recommendation
>
> Use the event stream service to receive the event stream of VM2000. For examples of the structure, see the “Introduction to System Administration” [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 interaction of BS2000 and VM2000 commands from a privileged user task on an SU /390.

```
/.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 '0CM4' WITH TSN '0CM5' ASSIGNED TO USER TASK '0CM4'
/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)  20.00  100.00  2(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 1 MONITOR   RUNNING         V21.0B  NO  YES   AG,AS
F R E E      VM: 14          LARGEST-DOMAIN(MB):   15344
/create-vm memory-size=512,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=990f,vm-id=system2,type=*sd -----------------------  (8)
VMS4005 DEVICE '990F' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (TYPE=SD)
/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=fc1a)) ---------- (10)
VMS4005 DEVICE 'FC1A' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'FC1B' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4220 PUBSET '4IVK' WITH PUBRES DEVICE 'FC1A' ASSIGNED TO VM (2,SYSTEM2)
/start-vm ipl-unit=fc1a,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-IO
 2 SYSTEM2   *STDPOOL(P)  30.00  100.00  1(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   RUNNING                 NO  NO    AS ------------------------ (13)
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    DC    NO    00(RUN),01(OFF) ---------------------------------- (14)
VM-ID        VM-MONJV
 2 SYSTEM2   NO VM-MONJV ASSIGNED ---------------------------------------- (15)
VM-ID        UNITS
 2 SYSTEM2   ASSIGNED BY VM-ADMIN ---------------------------------------- (16)
             EXCL:   CH,CI,FC1A,FC1B
```

```
             SHARED: 990F
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 2 SYSTEM2   00000210        512        512 ------------------------------ (17)
VM-ID        PRIV    DETAILS
 2 SYSTEM2   AS ------------------------------------------------------------ (18)
/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)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   RUNNING         V20.0B  NO  NO    AS ------------------------- (19)
/shutdown-vm vm-id=system2,action=*shutdown(quiet-time=0) ----------------- (20)
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (2,SYSTEM2) INITIATED
/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)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   DOWN            V20.0B  NO  NO    AS ------------------------- (21)
/delete-vm vm-id=system2 ------------------------------------------------- (22)
VMS4009 DEVICE 'CH' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'CI' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE '990F' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'FC1A' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'FC1B' 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 ------------------------------------------------------- (23)
```

```
   File   Edit   View                                                  Help
--------------------------------------------------------------------------------
                          EVENT STREAM SERVICE
 Log-Id: OBMU      User-Id: TSOS          Owner-TSN: OBMU          Mode: Static
--------------------------------------------------------------------------------
 Date: <date>       Time: <time>                                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 ...
```

```
/end-vm-dialog------------------------------------------------------------------------ (24)
/assign-sysevent to=*dummy------------------------------------------------------------ (25)
/assign-syslst to=*primary
/print-document from-file=list.vm2adm------------------------------------------------- (26)
/logoff system-output=*delete--------------------------------------------------------- (27)

-------------------------------------------------------------------------------------------------------------
```

(1) Login with 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. From this time, asynchronous messages from VM2000 to the VM2000 administrator are logged 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.

(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 BS2000 version in the guest system is (still) unknown.

(14) The virtual CPU `00` of VM2 is running (fixed CPU assignment to a real CPU). The virtual spare CPU `01` is not ready (offline). Later it will be placed in a ready status (SLEEP) by the mono-guest system.

(15) No MONJV is assigned to the VM.

(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) The main memory size and the location of the VM in the VM2000 main memory are output.

(18) The VM has the (default) privilege `AUTO-SNAP-ASSIGNMENT`.

(19) The guest system on VM2 has now reached `SYSTEM READY`. VM2000 recognizes the version of the BS2000 guest system.

(20) The VM2000 administrator initiates the shutdown for the guest system on VM2 without a QUIET time.

(21) After shutdown, VM2 is in the `DOWN` state.

(22) VM2 is terminated by the VM2000 administrator.

(23) The presentation function of the event stream service (ESS) is called. The asynchronous messages which have arrived for the VM2000 administrator are displayed.

(24) The VM2000 administrator terminates his/her ADMIN dialog with the monitor VM.

(25) The assignment of the user event stream to the user task is canceled.

(26) The SYSLST assignment is canceled and the session log is printed out.

(27) 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 an emulated console screen is meant.

Where necessary, BS2000 consoles are subdivided into:

- **KVP consoles** (see "Operating the guest systems using KVP consoles")
  The operator obtains access to the operator functions of a guest system through the emulation of a console screen on the SE Manager by the KVP (console distribution program).

  KVP consoles can be used in VM2000 operation:

  - for IPL, 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
- **Logical consoles** (see "Operating the guest systems using logical consoles")

  Logical consoles can be used in VM2000 operation:

  - for operation in the monitor system and in the other guest systems when BCAM is active in the system concerned
  - to operate the VM2000 system as a whole, when BCAM is active in the monitor system

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

  Authorization code “9” is required for operating the whole VM2000 system 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.
- **VC dialog via $VMCONS** (see "Operating the guest systems (VC dialog via $VMCONS)")
  This operating method is **no longer** recommended.

  Virtual consoles can be used in VM2000 operation:

  - for operation in the guest systems when BCAM is active in the monitor system, but not for the monitor system.

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.

5.1.3.1 Operating the guest systems using KVP consoles

Guest system operation with KVP consoles is controlled by console distribution programs (KVPs). KVP devices are configured on SU x86 or directly on the MU (for SU /390) using the SE Manager when the system is installed and during VM2000 installation, see the “Operation and Administration” manual [19]. The console screens are integrated into the SE manager.

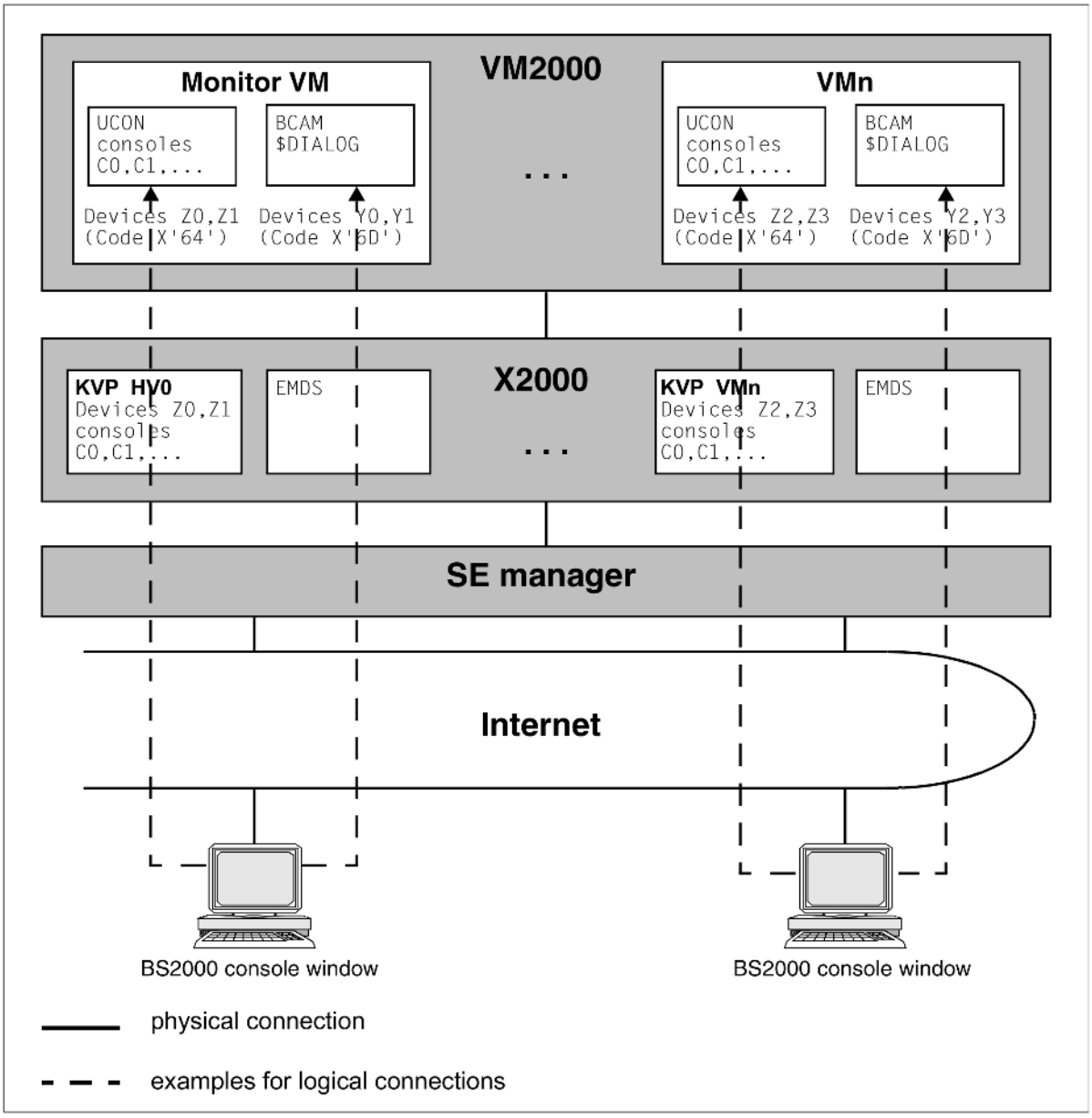


Figure 7: Example of communication with guest systems via KVP consoles (SU x86)

Preparations

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

1. A separate KVP must be created via the SE manager for every guest system to be operated. Each of these KVPs is represented, with respect to VM2000 and the BS2000 guest system, by its two mnemonic device names (KVP console device pair).

   Such device pairs can also be added to the system during ongoing operation, see "Changing the configuration of devices via X2000 (SU x86)". 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.
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 SE Manager and opening a BS2000 console window in the *Operation* tab.

   After logging in to a console, the KVP reports to the BS2000 guest system that this console is operable. Several guest system operators can open their own console windows with the same BS2000 console; 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 "START-VM (Start a guest system on a VM)".

Remote Service

Only Customer Service ever makes settings for Remote Service. Service access can be set up for every VM, see the manual “Operation and Administration” [19].

5.1.3.2 Operating the guest systems using logical consoles

If BCAM is 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. When BCAM is 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 [12]). 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
  When BCAM is 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”).

5.1.4 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 carried out from a privileged user task in the monitor system (see "Privileges of the user task in the monitor system").
> The guest system on a VM is normally operated with KVP consoles or logical consoles (see "Operating the guest systems using BS2000 consoles").

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 when BCAM is active 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 "Setting up the VM2000 monitor"). 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 "MODIFY-VMCONS-PARAMETERS (Modify parameters of $VMCONS)".

> 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 "Setting up the VM2000 monitor") should also be updated.

$VMCONS connects to the $CONSOLE application as an authorized user program with a dynamic authorization name (see the “Introduction to System Administration” [2]).
An operator ID (`VMCONS-OPERID`) and an operator role (`VMCONS-OPERATOR-ROLE`) must be defined for running $VMCONS (see "Preparing to use $VMCONS").

> ¡ 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 “Introduction to System Administration” [2]) allows 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 will usually be used for access from a terminal.

When BCAM is terminated in the monitor system, $VMCONS is also terminated. If $VMCONS terminates abnormally, it is restarted automatically (see "Restarting a task after an error in a system task of VM2000"). 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 will usually be used for access from 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").
>
> 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 [,<proname> 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 [12]):

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

<pac>	Freely-selectable OMNIS connection name.
PRONAME=	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 "Setting the VM2000 parameters in the parameter service").

> `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 "Operating the guest systems (VC dialog via $VMCONS)".

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 a connection name prefixing 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 SE manager).
>
> 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 "Operating the virtual machines (ADMIN dialog via $VMCONS)") or `GS02`, `GS03`, ..., `GSnn` if a VC dialog is to be conducted via this connection (see "Operating the guest systems (VC dialog via $VMCONS)").

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 "Example of ADMIN and VC dialog via $VMCONS" 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 "Privileges of the user task in the monitor system").

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

- Establish a connection to $VMCONS via OMNIS (see "Establishing a connection to $VMCONS") 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 of the monitor system under this authorization name (`@nnn, n=0..9`).
- Enter VM2000 commands to operate the VM (see "Operating a VM" and "VM2000 commands") or the VM2000 system as a whole (see "Operating the whole VM2000 system" and "VM2000 commands").
- 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 "Establishing a connection to $VMCONS"). 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 "Establishing a connection to $VMCONS".

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 [12]). When OMNIS logging is activated, OMNIS logs all messages 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 commands and 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 "Leading a VC dialog").

The guest system can be started up and operated via the VC dialog. An example of this is provided in the section "Example of ADMIN and VC dialog via $VMCONS".

> ¡ We recommend operating the guest systems using KVP consoles or logical consoles (see "Operating the guest systems using BS2000 consoles") instead of via $VMCONS.

Implementing a VC dialog via $VMCONS connection to KVP (SU x86)

A VC dialog on SU x86 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 (SU /390)

A **virtual console** is a device that is generated as a physical console at hardware generation of the SU /390 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 "Leading a VC dialog").

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 is provided in the section "Example of ADMIN and VC dialog via $VMCONS". You cannot use a virtual console for the monitor system.

> ¡ We recommend operating the guest systems using KVP consoles or logical consoles (see "Operating the guest systems using BS2000 consoles") instead of via $VMCONS.

Generating the virtual console

Virtual consoles must be generated as devices, see "Device peripherals on SU /390".

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.

Leading 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 "Establishing a connection to $VMCONS") 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"".

- 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 only VC dialogs are carried out 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 carried out in parallel with one guest system. Several operating or monitoring instances can thus operate the guest system in parallel. Please observe the *Note* in the section "Establishing a connection to $VMCONS".

All the VC dialogs of a guest system have the same command and message authorization. When BCAM is 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 "Restart routines in VM2000")
- 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.

390 On SU /390, 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).

x86 Outputs of the guest systems are not buffered on SU x86 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 (SU /390)

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 8), this places an additional burden on the VM2000 hypervisor and the monitor VM.

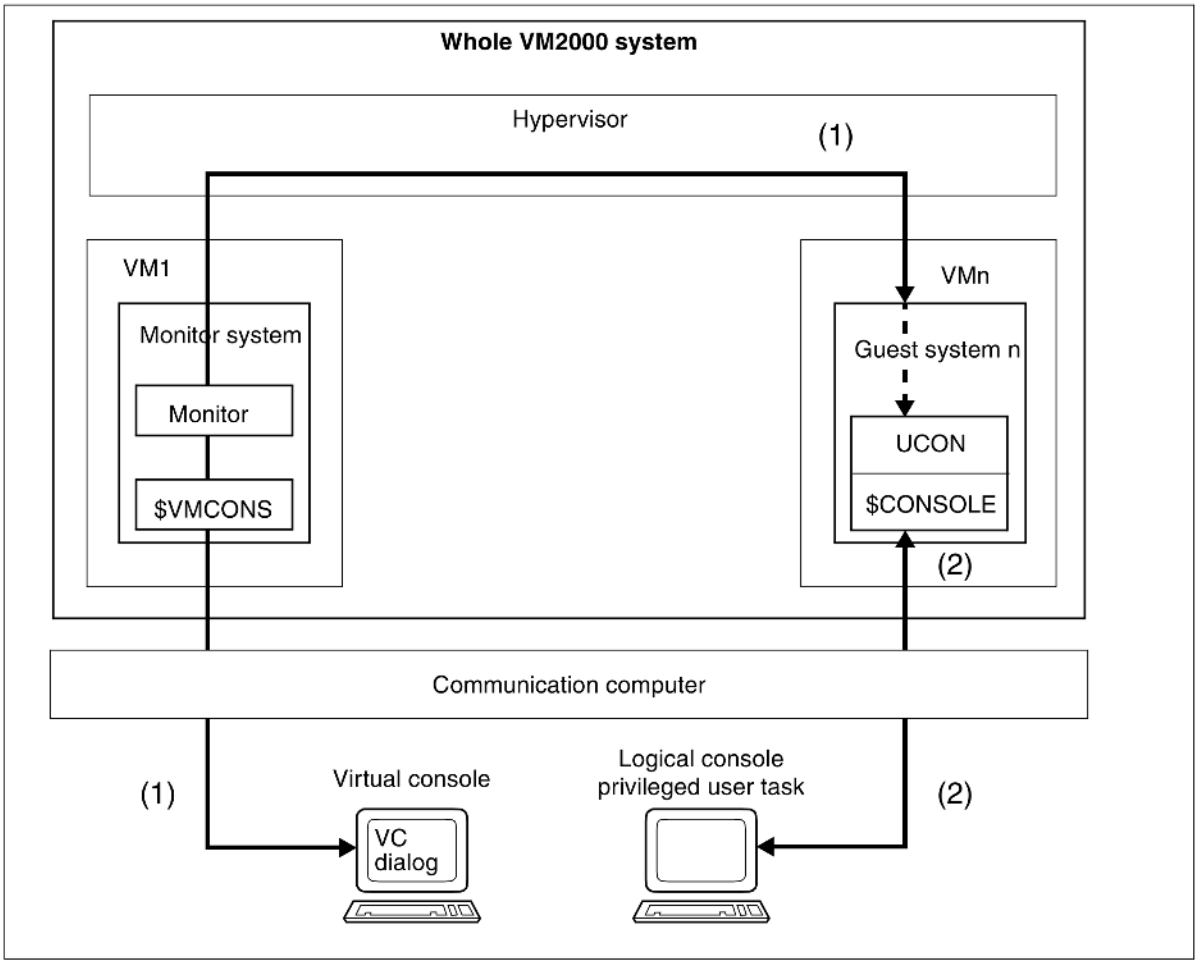


Figure 8: Message flow when operating the guest system

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) when BCAM is active in the guest system (see the “OMNIS/OMNIS-MENU” manual [12]).

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 8). 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 manual “OMNIS/OMNIS-MENU” [12].

The connection to $VMCONS is established, as described in the section "Establishing a connection to $VMCONS", 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 in the section "Establishing a connection to $VMCONS").
> - **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) on an SU /390.

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 "Example of an ADMIN dialog from a user task".

Tracer listing

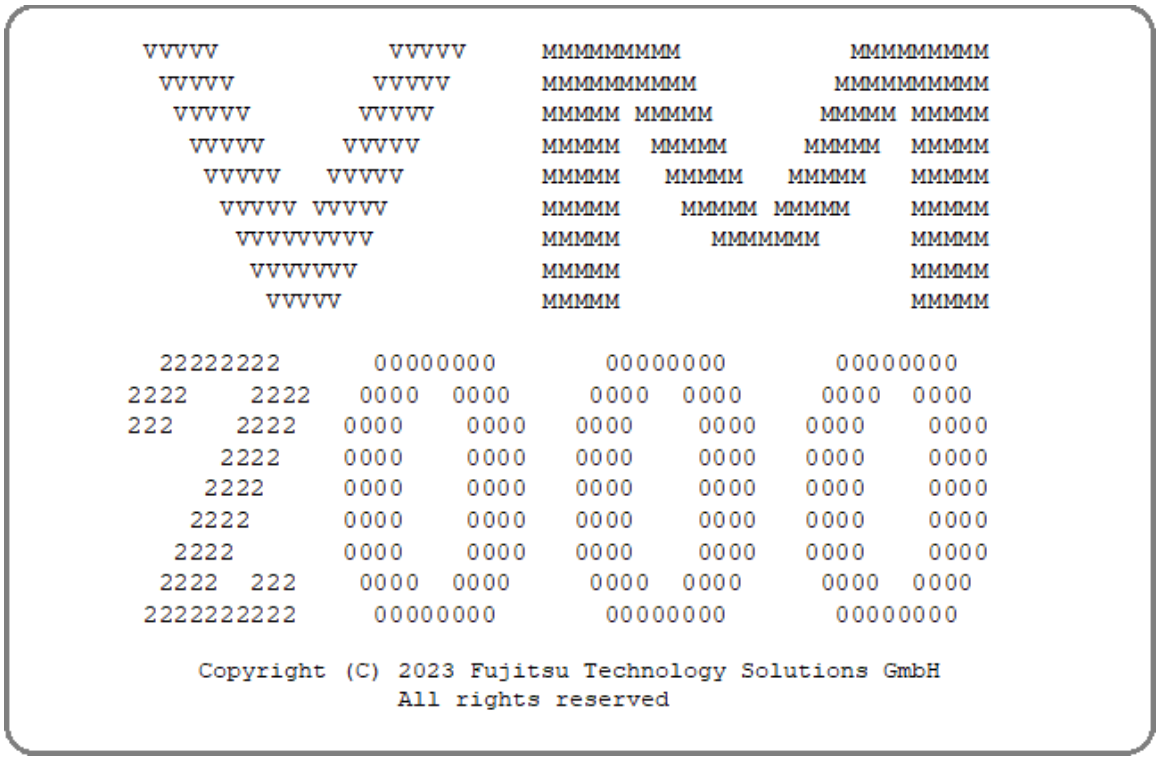
```
@: OMS0001 USER IS CONNECTED TO OMNIS (TERMINAL=(STATFD70/MCH9700D);
   TID=   474)
@: OMS0002 OMNIS VERSION <ver> READY ————————————————————————————  (1)
@:option logging=yes ————————————————————————————————————————————  (2)
@:option input-logging=send,output-logging=send
@:option pac=std ————————————————————————————————————————————————  (3)
@:option break-key=1,call-inf=yes,call-key=2 ————————————————————  (4)
@:opncon vm01,typ=dcam,ptnname=$vmcons,proname=d021ze01,appname=OMS111   (5)
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=VM01, PID=   477,
           APPNAME=(OMS111/D021ZE01))
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY
           (PARTNER=($VMCONS/D021ZE01), PAC=VM01) ———————————————————  (6)
```

```
 VVVVV           VVVVV     MMMMMMMMM           MMMMMMMMM
  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
       VVVVVVVVV           MMMMM     MMMMMMM      MMMMM
        VVVVVVV            MMMMM                  MMMMM
         VVVVV             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
      2222      0000    0000   0000    0000   0000    0000
     2222       0000    0000   0000    0000   0000    0000
    2222        0000    0000   0000    0000   0000    0000
   2222  222     0000  0000     0000  0000     0000  0000
  2222222222     00000000       00000000       00000000

    Copyright (C) 2023 Fujitsu Technology Solutions GmbH
                  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: % NBR0740 COMMAND COMPLETED 'BEGIN-VM-DIALOG';
                (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date> --  (8)
/create-vm memory-size=512,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 '*STDPOOL'
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=fc1a)) --------- (10)
VM01: % VMS4005 DEVICE 'FC1A' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VM01: % VMS4005 DEVICE 'FC1B' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VM01: % VMS4220 PUBSET '4IVK' WITH PUBRES DEVICE 'FC1A'
                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 units=(cc),type=*vc,vm-id=system2 ------------------------ (11)
VM01: % VMS4005 DEVICE 'CC' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
                 (TYPE=VC)
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>
@:/opncon gs02,typ=ucon,ptnname=$vmcons,proname=D021ZE01,appname=OMS112 (12)
@: OMS0012 'OPNCON' COMMAND ACCEPTED (PAC=GS02, PID=   487,
           APPNAME=(OMS112/D021ZE01))
@: OMS0014 'OPNCON' COMMAND EXECUTED SUCCESSFULLY
           (PARTNER=($VMCONS/D021ZE01), PAC=GS02)
```




```
gs02:/begin-vm-dialog vm-id=system2,password=c'sys2',
                      type=*vc(output-prefix=*no)  ———————————————————— (13)
VM01: % VMS1033 PARTNER 'OMS112   D016ZE04' 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';
                (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
vm01:/start-vm ipl-unit=fc1a,vm-id=system2,information-byte=*dialog —— (15)
VM01: % VMS2023 CPU 00 OF VIRTUAL MACHINE (2,SYSTEM2) STARTED
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.NSI00E3 IPL-REPS READ: 2; EXECUTED: 2
GS02: %P.NSI1100 IPL DEVICE = 4IVK.0; IPL PATH = FC1A (MN=FC1A)
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 002 PUBVOLS OF HOME-PVS 4IVK WITH SYSID 186 ONLINE
GS02: %P.NSI1145 4IVK.0 <date> <time> ON FC1A
GS02: %P.NSI1145 4IVK.1 <date> <time> ON FC1B
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> = 0076A000
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)
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 ($VMCONS/D021ZE01) LOST
           (PAC=GS02; CODE=00000000) ------------------------------------- (21)
vm01:/end-vm-dialog  ------------------------------------------------------ (22)
VM01: % VMS3000 VM2000 COMMAND /END-VM-DIALOG FOR
                VIRTUAL MACHINE (1,MONITOR) PROCESSED SUCCESSFULLY
VM01: % NBR0740 COMMAND COMPLETED 'END-VM-DIALOG';
                (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
@: OMS0006 CONNECTION TO PARTNER ($VMCONS/D021ZE01) LOST
           (PAC=VM01;CODE=00000000)
```

---

(1) Via a terminal emulation the connection is established to OMNIS.

(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 in 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.

(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.

(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 a virtual console to VM2.

(12) A further connection to $VMCONS is established via OMNIS with `TYP=UCON` and 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.

(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.

(19) The VC dialog is terminated. The OMNIS connection name `GS02` would only have to be specified if the OMNIS connection were changed. Specification of `VM00` as a replacement prefix is necessary, as only a VC dialog (`TYPE=*VC`) is carried out 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 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
  - guaranteed messages
  - SHOW output in structured S variables

  For an example of an S procedure under VM2000, see below. 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)". 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 (Start a VM2000 procedure file)".
  VM2000 procedures contain VM2000 commands.
- Persistent VM definitions
  A persistent VM definition is also available to a Server Unit after a reboot. With its help a persistent VM is set up again and restarted immediately when the corresponding specification is provided in the `AUTO-IPL` parameter. See section "Working with VM definitions".
- 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 "Backing up and restoring the VM2000 configuration".

Monitoring with job variables

The routine monitoring of virtual machines, of main memory reconfigurations and of VM Migrations can be automated under VM2000 by using job variables (MONJVs), see "Monitoring a VM with job variables (MONJV)", "Monitoring a VM Migration with job variables (MONJV)" and "Reconfiguring main memory". 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   VMS2411 -
VMS2422   VMS2424   VMS2425   VMS2430 - VMS2434   VMS2440 - VMS2444   VMS2450 -
VMS2452   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   VMS4910
VMS4912   VMS6002   VMS6003   VMS6018   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 [12].

  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 [15].

  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.

  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 "Reconfiguring main memory".

```
/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                        "
/" ** - CONNECTION TO VM2000 ESTABLISHED                       "
/" ** - 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                        "
/" ** - DISCONNECTION FROM 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.I0='2'))
/  CONT=FALSE
/ ELSE-IF ((SYSPOP.EVENT-NAME = 'VMS4095') AND (SYSPOP.I0 = '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.I0='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 list of VM2000 commands and their function groups is provided in section "Grouping of VM2000 commands according to function". VM2000 commands are logged, see "Administration dialogs from privileged user tasks" and "Operating the virtual machines (ADMIN dialog via $VMCONS)".

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 **states**:

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.
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 is terminated.

These states are brought about by the following VM2000 commands or events. The commands are described in detail in the following sections.

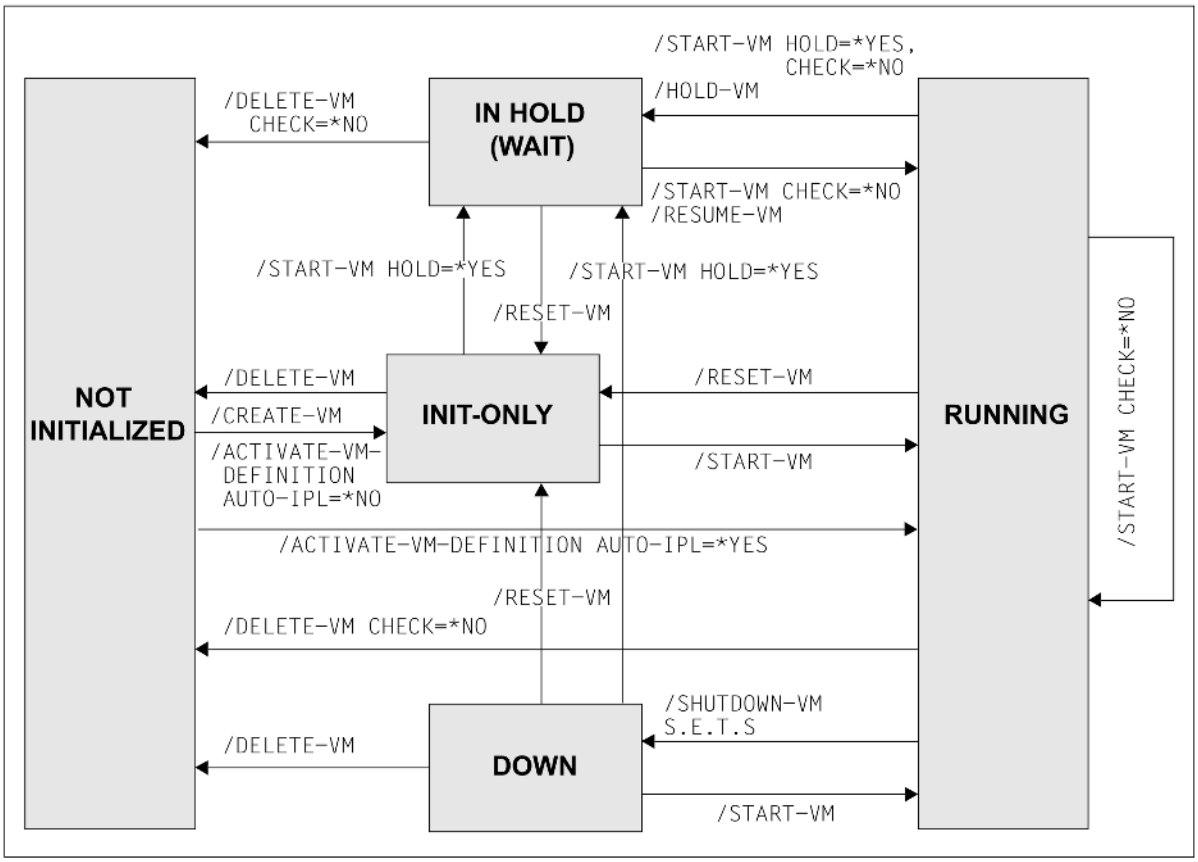


Figure 9: State diagram of a VM

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` (VM2000 administrator). This requires that no VM or VM definition has been configured with the specified VM name. The VM administrator is allowed to terminate but not to initialize a VM.

> ¡ A VM can also be initialized with `/ACTIVATE-VM-DEFINITION` and possibly restarted if a corresponding persistent VM definition exists for this purpose. Further information on VM definitions is provided in the section "Working with VM definitions".

During the initialization process, 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 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
- Password for dialog access
- Range of commands available to VM2000 and VM administrators
- Privileges of the VM
- Settings for control over the real CPU
- PERSISTENT attribute

After the successful initialization, the VM has the status INIT-ONLY.

x86 On SU x86, 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").

The maximum number of VMs which can be initialized depends on the architecture of the Server Unit, see "CREATE-VM (Initialize a VM)". It is also displayed when `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION` is used.

Identification of the VM

The identification **VM-ID** identifies the VM in the VM2000 commands. The VM-ID can be the **VM index** or the **VM name**. VM index and VM name are assigned to a VM during initialization. These identify a VM unambiguously and can no longer be changed following initialization of the VM.

The VM index is an integer n from 1 to 99 (the upper limit depends on the architecture of the Server Unit) and identifies the VM (VM1 to VMn). 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). The name should reflect the user or the usage mode of the VM. It should be specified explicitly to avoid standard names which are not very meaningful.
If no VM name is specified, VM2000 assigns the standard name `VM00nn`, where `nn` is the VM index (`nn=01..99`). Initialization of the 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.

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.

x86 The default name of the monitor VM on SU x86 is `MONITOR`.

When a VM is configured on SU x86, the VM name is used as the domain name.
Therefore, the special characters **#**, **$** and **@** should not be used in the VM name; they are replaced by **n**, **s** and **a** in the domain name.

The name ranges for VMs, VM groups and CPU pools should be disjunctive.

> **i** 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 "Accounting in VM2000").

Size of VM main memory

This attribute determines the main memory size for the VM (see section "Managing main memory"). The maximum main memory size under VM2000 is 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

390 On SU /390, a main memory area begins on a 1-Mbyte boundary and its size is a multiple of 1 Mbyte.

x86 On SU x86, 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 SU x86 must therefore be at least 1024 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 is to be reduced while the guest system is active (see "Reconfiguring main memory").

The minimum main memory size can be increased with /EXTEND-VM-MEMORY. On SU /390 it can be decreased (implicitly) with /REDUCE-VM-MEMORY (see "REDUCE-VM-MEMORY (Reduce main memory for a VM)").

> **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 [9]): `Resident Memory = TOTAL - Pageable Frames .`
>
> x86 On SU x86, the minimum size of a VM’s main memory must be at least 1024 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 (SU x86)

The maximum size of the main memory should be defined for a VM only when the main memory of the VM is to be extended while the guest system is active (see "Reconfiguring main memory").

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 value combination is rejected (VMS4093).

390 On SU /390, 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

390 On SU /390, this attribute determines the location of the VM in the main memory of VM2000 (see "Main memory management and reconfiguration"). The address must be a multiple of 1 Mbyte. If the location 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.

x86 On SU x86, the location of a VM cannot be determined. 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 avaiable CPU capacity on the VMs.

390 On SU /390, 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.

x86 On SU x86, 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".

The CPU quota and maximum CPU utilization can be modified by means of /MODIFY-VM-ATTRIBUTES.

Maximum I/O utilization of the VM

The I/O utilization of a VM can be limited by the maximum I/O utilization of the VM.

390 On SU /390, the BS2000 subsystem IORM monitors the maximum I/O utilization in the IOLVM function, see "Use of IORM in VM2000 operation".

x86 On SU x86, only the default value (100, unlimited utilization) can be used for this attribute.

The maximum I/O utilization can be changed using /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.

390 On SU /390, 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.

x86 VM groups are not available on SU x86.

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, (SU /390, 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".

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)
. . .		. . .
32	(32-Way)	32 processors (virtual CPUs 0 through 31)

390 On SU /390, the maximum multiprocessor level is 16.

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 in the section "Increasing capacity with extra CPUs".

The virtual CPUs of the VMs that result from this are started up on the available real CPUs, see section "Scheduling procedure".

The multiprocessor level of a VM constitutes the upper limit for the maximum CPU utilization, see "CPU quota and maximum CPU utilization of the VM". For example, a biprocessor-VM can accommodate the CPU capacity of up to two real CPUs.

The multiprocessor level of the monitor VM is set when VM2000 is installed (see chapter "Installing VM2000").

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 "Extending and restricting the range of commands/functions".

The range of commands can subsequently be modified by means of `/MODIFY-VM-ATTRIBUTES`.

Privileges of the VM

IO-RESET privilege

The I/O 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` (when initializing the VM or `/MODIFY-VM-ATTRIBUTES`).

390 On SU /390, 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.

x86 On SU x86, 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 SU /390:

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 in 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 during its initialization or with /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 "Scheduling procedure"). 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 SU /390 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 when the VM is initialized or with /MODIFY-VM-ATTRIBUTES. It applies to all virtual CPUs of the VM.

390 On SU /390, 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.

x86 On SU x86, only the default value (*NO, no I/O 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 Unit.

AUTO-SNAP-ASSIGNMENT privilege

This privilege permits the guest system on a VM to assign itself snap units of a Snapset 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 "Assignment sets, implicit device assignment and release".

To permit implicit device assignment, the VM must have the ASSIGN-BY-GUEST privilege for the required assignment sets (when the VM is initialized or later with /MODIFY-VM-ATTRIBUTES).

Each device that is to be implicitly assigned must also have the attribute ASSIGN-BY-GUEST (see /MODIFY-VM-DEVICE-ATTRIBUTES). The device is also assigned to the required assignment set here.

Settings for control over the real CPU

On SU /390 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 IDLE).

x86 On SU x86, only the default value (*NO) can be used for this attribute.

When VM-ACTIVE-IDLE=*NO (on SU /390), the VM leaves control over the real CPU if the VM's virtual CPU which runs on this becomes 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 becomes 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 a virtual console.

> ¡ With fixed CPU assignment, VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS is only effective if the VM's maximum CPU utilization (see "CPU quota and maximum CPU utilization of the VM") is not restricted.

The setting can also be modified later using /MODIFY-VM-ATTRIBUTES.

PERSISTENT attribute

The PERSISTENT attribute defines whether a persistent VM is created (PERSISTENT=*YES). A persistent VM is assigned a persistent VM definition.

The PERSISTENT attribute **cannot** be set for the monitor VM.

A persistent VM definition is also available to a Server Unit after a reboot. With its help a persistent VM is set up again and restarted immediately when the corresponding specification is provided in the AUTO-IPL parameter. A persistent VM definition is not deleted with /DELETE-VM.

A VM without the PERSISTENT attribute is assigned a non-persistent VM definition. This is deleted with /DELETE-VM.

Further information on VM definitions is provided in the section "Working with VM definitions".

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.

In 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 (see "Initializing a VM"), a monitoring job variable (MONJV) is defined for this purpose. 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.
> VMs initialized with `/ACTIVATE-VM-DEFINITION` can also be monitored with job variables. Detailed information can be found in the section "Working with VM definitions".

A prerequisite for working with job variables is the software product JV. For more information on job variables, see the manual “Job Variables” [17], especially the section “Monitoring jobs/programs with job variables”.

For an example of how to use a MONJV to monitor a VM, see the end of this section.

MONJVs can also be used to monitor VM Migrations and main memory reconfigurations.

Beginning and end of monitoring

The MONJV used to monitor the VM is configured when the VM is initialized (see "Initializing a VM") by specifying the `MONJV` operand. 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 and with a password, this password can be specified in the operand `JV-PASSWORD` to check for access authorization.

The monitoring ends as soon as the VM is terminated, after migrating the VM or when VM2000 is terminated (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 `$I`) 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 for any user ID they wish. Nonprivileged users can create the MONJV only for their own user ID.
If catalog and user IDs are not specified, the MONJV of VM2000 will be created (shared) for the user ID of the sender of the command and 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 for TSOS.
If catalog and user IDs are not specified, the MONJV of VM2000 will be created (shared) for TSOS on the home pubset of the monitor system and 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 "Setting up the VM2000 monitor").
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 (left-aligned, padded with a space): • $S : VM initialization initiated (Creation Started) • $A : VM initialization terminated with error (Creation Aborted) • $I : VM initialized (VM status INIT-ONLY) • $R : VM running (VM status RUNNING) • $D : guest system terminated on the VM (VM status DOWN) • $H : VM stopped (VM status IN HOLD (WAIT)) • $T : VM terminated (Terminated, VM status NOT INITIALIZED)
4	Reserved, has the value zero (0)
5-8	Reserved, contains spaces
9-12	Catalog ID of the home pubset of the monitor system (left-aligned)
13-16	Reserved, contains spaces
17	Type of MONJV: character V (VM2000)
18-20	Current system sequence number (<integer 1..999>)
21-36	Beginning of monitoring (opening the MONJV) in UTC time (format: yyyy-mm-ddhhmmss)
37-70	Reserved
71-78	Name of VM (<name 1..8>, not for status$S and $A)
79-81	Index of VM printable (<name 3..3>, e.g. 002)
82-86	Status of guest system: • 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)
87-94	Migration status of the VM: • MIGR-OUT: The VM is being migrated to another Server Unit
95-128	Reserved
129-256	Reserved for users

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 [17]).

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 during Migration of the VM

MONJVs used to monitor VMs are **not** migrated to the target SU during a VM Migration. After the VM has been terminated on the local Server Unit, the MONJV is retained as a user job variable. After successful migration of the VM to the target SU, the MONJV is set to monitoring state `$T` (VM terminated).

A migrated VM cannot be monitored with a MONJV.

Behavior when the system is terminated abnormally in the monitor system

MONJVs are retained in the monitor system when the system has been 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 system or a VM outside the monitor VM, the program 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 [10], 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.

In a later session this MONJV can be used again to monitor a newly intialized VM (see "Initializing 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 BS2000 console of the monitor system.

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'
```

```
/ 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 (see "Initializing a VM"), it can be assigned pubsets and other devices. To permit implicit assignment of pubsets 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".

> ℹ `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used to enter devices and assignment sets for the VM in a VM definition. Further information on VM definitions is provided in the section "Working with VM definitions".

5.2.4 VM and VM group

x86 VM groups are not available on SU x86.

On SU /390, VMs are added to a VM group by the VM2000 administrator when the VM is created (see "Initializing a VM") or during ongoing operation using `/ADD-VM-TO-VM-GROUP`, see the section "Adding a VM to a VM group".

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".

> `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used to enter the affiliation of a VM to a VM group in a VM definition. Detailed information can be found in the section "Working with VM definitions".

5.2.5 Assigning a VM or a VM group to a CPU pool

VMs which do not belong to a VM group are assigned to a CPU pool when the VM is initialized, see "Initializing a VM". 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".

> ¡ `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used in a VM definition to enter the affiliation of a VM to a CPU pool in a VM definition. Detailed information can be found in the section "Working with VM definitions".

On SU /390, the VM2000 administrator assigns VM groups to a CPU pool when the VM group is initialized using `/CREATE-VM-GROUP` (`CPU-POOL` operand), see "Creating a VM group". 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".

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".

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 carried out 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 (see "Initializing a 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 “Introduction to System Administration” [2]).

> ¡ How the monitor system is started is described in the chapter "Initializing and terminating VM2000".
> `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used in a VM definition to enter details for the automatic start of the guest system when activating the VM definition with `/ACTIVATE-VM-DEFINITION`. Detailed information can be found in the section "Working with VM definitions".

x86 On SU x86, the guest system can also be started using the SVP functions of the KVP console which is assigned to the VM (see the manual “Operation and Administration” [19]).

Before starting the guest system on a VM, at least one console device (a KVP console device pair or, on SU /390 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 "Assignment sets, implicit device assignment and release")

In order to start the guest system on the VM, the device with the IPL disk device must be specified at the **first** start.

> ¡ This does not apply for the first startup after `/ACTIVATE-VM-DEFINITION` (without specifying AUTO-IPL) if the last IPL device used is entered in the VM definition. The last IPL device used is employed.

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 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 is not specified the first time the guest system is started on the VM, VM2000 searches for a KVP console device pair assigned to the VM.

  > This does not apply for the first startup after /ACTIVATE-VM-DEFINITION if the last IPL console used is entered in the VM definition. The last IPL console used is employed.

  390 If, on SU /390, no KVP console device pair is assigned to the VM, the virtual console of the VM is used. The virtual console is also used if a KVP device pair is assigned to the VM but the mnemonic device name of the virtual console was specified explicitly for /START-VM or it is entered as the last IPL console 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).
- 390 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.
- 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.

390 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 SU x86, the system name (= VM name) are retained for the next startup of the guest system until a VM is shut down or until the VM definition has been deleted. They provide the default values for the next startup.

State of the VM

VM2000 prevents the accidental starting of a VM on which the guest system is still active (CHECK-VM-STATE=*YES).
If the guest system is to be started on a VM in the IN HOLD (WAIT) or RUNNING state, 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)
  x86 Only the default value HOLD=*NO is allowed on SU x86.
- IN HOLD (WAIT) (operand HOLD=*YES)

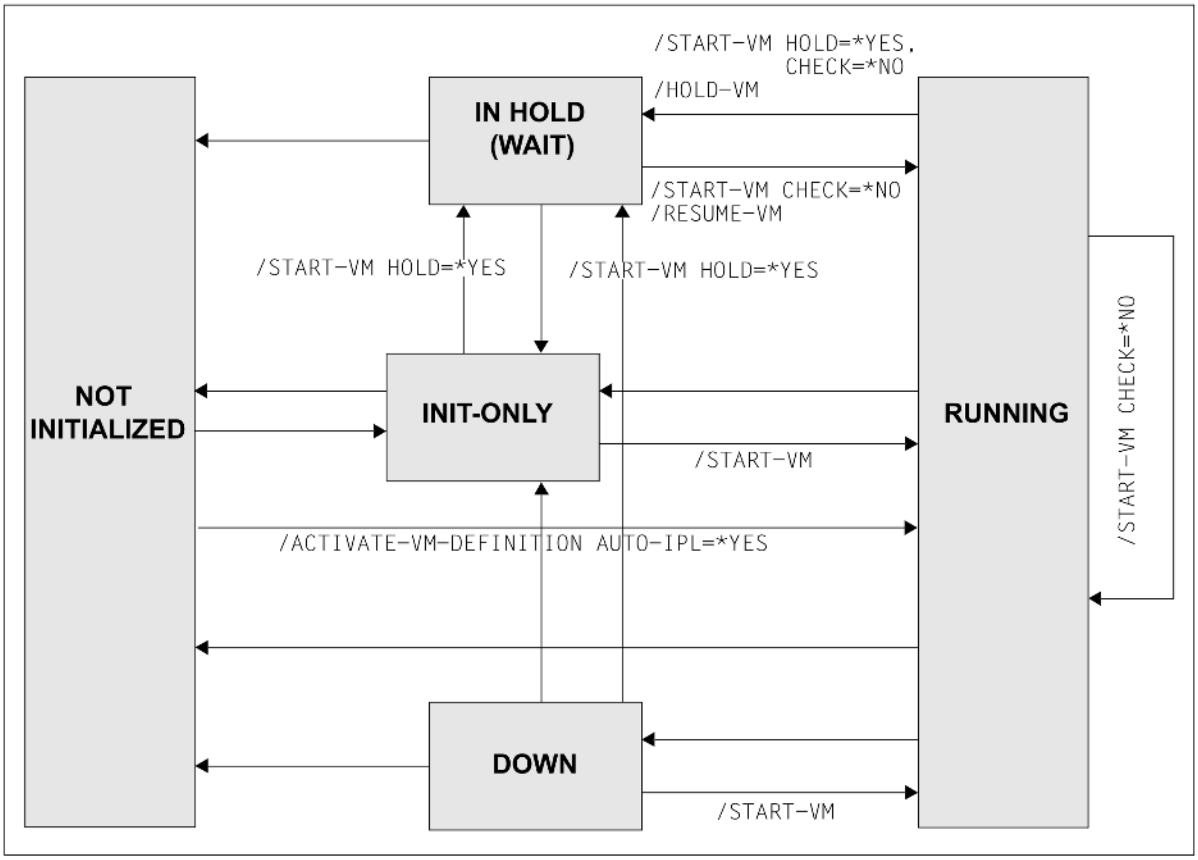


Figure 10: Changing the VM state by means of `/START-VM` and `/ACTIVATE-VM-DEFINITION`

5.2.8 Information services

The information services supply information

- for the VM2000 administrator
  - on the distribution of VM2000 resources (/SHOW-VM-RESOURCES)
  - the device status (/SHOW-VM-DEVICE-STATUS)
  - on the assignment sets (/SHOW-VM-ASSIGNMENT-SET)
  - on 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)
  - on the attributes of VM definitions (/SHOW-VM-DEFINITION)
- for the privileged VM administrator
  - on the distribution of VM2000 resources (/SHOW-VM-RESOURCES)
  - 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 within 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 Unit are shown with /SHOW-VM-RESOURCES, see "SHOW-VM-RESOURCES (Output status of VM2000 resources)".

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 "SHOW-VM-DEVICE-STATUS (Output detailed device information)".

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 "SHOW-VM-ASSIGNMENT-SET (Output assignment sets)".

This command is reserved 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 "SHOW-VM-STATUS (Output VM2000 monitored data)".

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 "SHOW-VM-CPU-POOL (Output attributes of a CPU pool)".

This command is reserved for the VM2000 administrator.

Information on the attributes of VM groups (SU /390)

/SHOW-VM-GROUP outputs detailed information on the attributes of the VM groups and the associated VMs, see "SHOW-VM-GROUP (Output attributes of VM group (SU /390))".

This command is reserved for the VM2000 administrator.

Information on VM definitions

/SHOW-VM-DEFINITION is used to display detailed information on the attributes, resources, devices and specifications for automatic start of the guest system on the VM which is entered for a VM in the VM definition, see "SHOW-VM-DEFINITION (Output attributes of a VM definition)".

This command is reserved for the VM2000 administrator.

Information on the attributes of a VM

The attributes of an initialized VM are output using /SHOW-VM-ATTRIBUTES, see "SHOW-VM-ATTRIBUTES (Output attributes of a VM)".

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 "Extending and restricting the range of commands/functions") can modify the attributes of a VM which were defined during VM initialization.

  The following changes can be made:

  - the CPU quota and the maximum CPU utilization
  - the maximum I/O 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 "Extending and restricting the range of commands /functions"
  - the privileges of the VM, see "Initializing a VM"
  - the setting for controlling the real CPU (SU /390)
  - the scope of the trace entries to be logged
  - the setting of the PERSISTENT attribute
  - the information for automatically starting the guest system on the VM when activating the VM definition
- of peripheral devices for a VM and of the VM2000 devices (see "Peripheral device management")

  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.
- of the main memory for the VM by reconfiguration (see "Reconfiguring main memory")
  The main memory for a VM can be enlarged with/EXTEND-VM-MEMORY and reduced with /REDUCE-VM-MEMORY.

  390 On SU /390, free main memory areas can also be extended or reduced by moving a VM with /MOVE-VM (see "Moving a VM within main memory").
- of real CPUs which are available for VM2000 operation (see "Reconfiguration of real and virtual CPUs")
  These can be attached using /ATTACH-VM-RESOURCES and detached using /DETACH-VM-RESOURCES.
- of the (dynamic) CPU pools (see "Managing CPU pools")
  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

On SU /390, a VM in the RUNNING state can be suspended using `/HOLD-VM`. `/RESUME-VM` cancels a previously initiated wait state.

x86 Commands `/HOLD-VM` and `/RESUME-VM` are not supported any more on SU x86.

> 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.

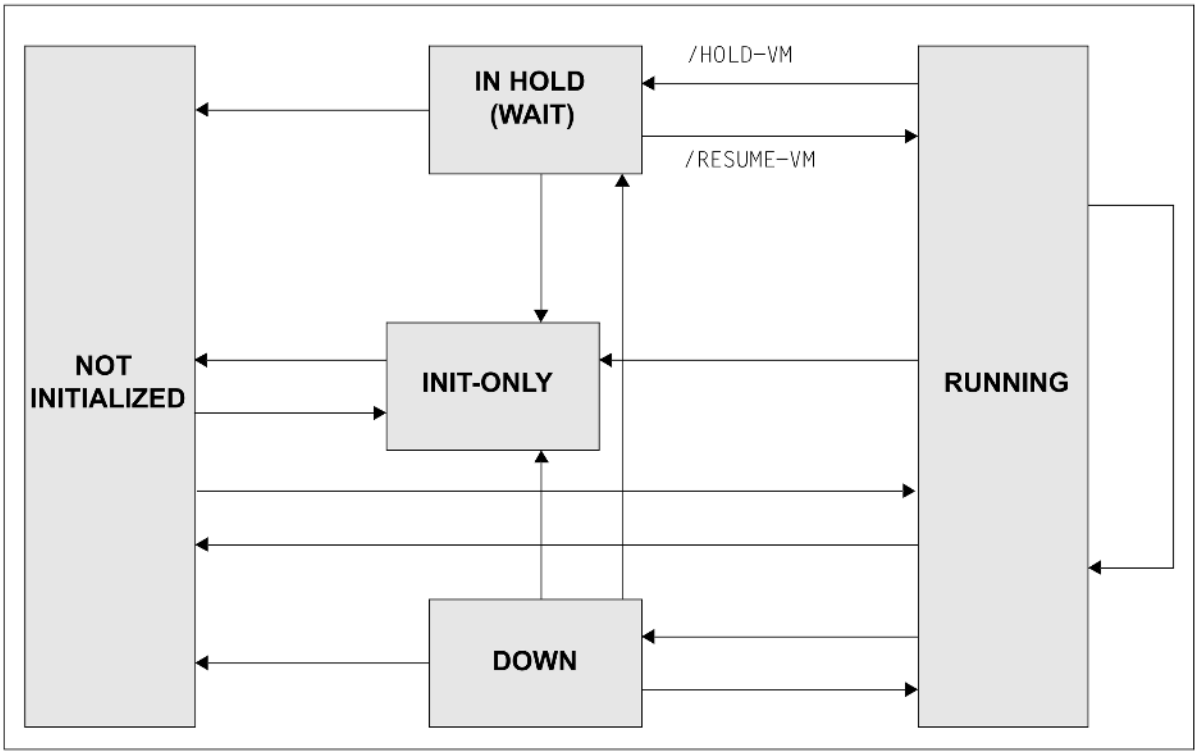


Figure 11: Changing the VM state by means of `/HOLD-VM` and `/RESUME-VM`

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. If there are overlapping wait states, only the wait state with the highest priority (in the order SEL, GLB, VMA) is displayed.

The following table shows the possible changes of state to which a VM can be subjected as a result of suspension.

Command /HOLD-VM **Old status**	**VM2000 administrator global (VM-ID=*ALL)**	**VM2000 administrator selective (VM-ID=2..n)**	**VM administrator (VM-ID=*CURRENT)**
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

Meaning:

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 Old status	VM2000 administrator global (VM-ID=*ALL)	VM2000 administrator selective (VM-ID=2..n)	VM administrator (VM-ID=*CURRENT)
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

x86 On SU x86, there is no need to change the location of a VM. /MOVE-VM is not available.

/MOVE-VM enables the VM2000 administrator on SU /390 to (retrospectively) change the location of a VM in main memory from the original location defined during initialization.

Overlapping with the old VM main memory is possible. The new address must be a multiple of 1 Mbyte.

It is possible to protect a guest system in the MSCF network from being regarded as failed by MSCF partners while it is being moved.

5.2.12 Sends a message 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 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 "Assignment sets, implicit device assignment and release") are, however, removed from the VM. If main memory reconfiguration if being carried out, it is terminated.

390 On SU /390 the Channel Subsystem Monitoring Facility function is switched off for the guest system.

The following tasks are facilitated with `/RESET-VM`:

- Final termination of a guest system which seems to have failed
- Immediate termination of a test system if shutdown is not wished for or not possible

State of the VM

After execution of `/RESET-VM`, the VM adopts the INIT-ONLY state.

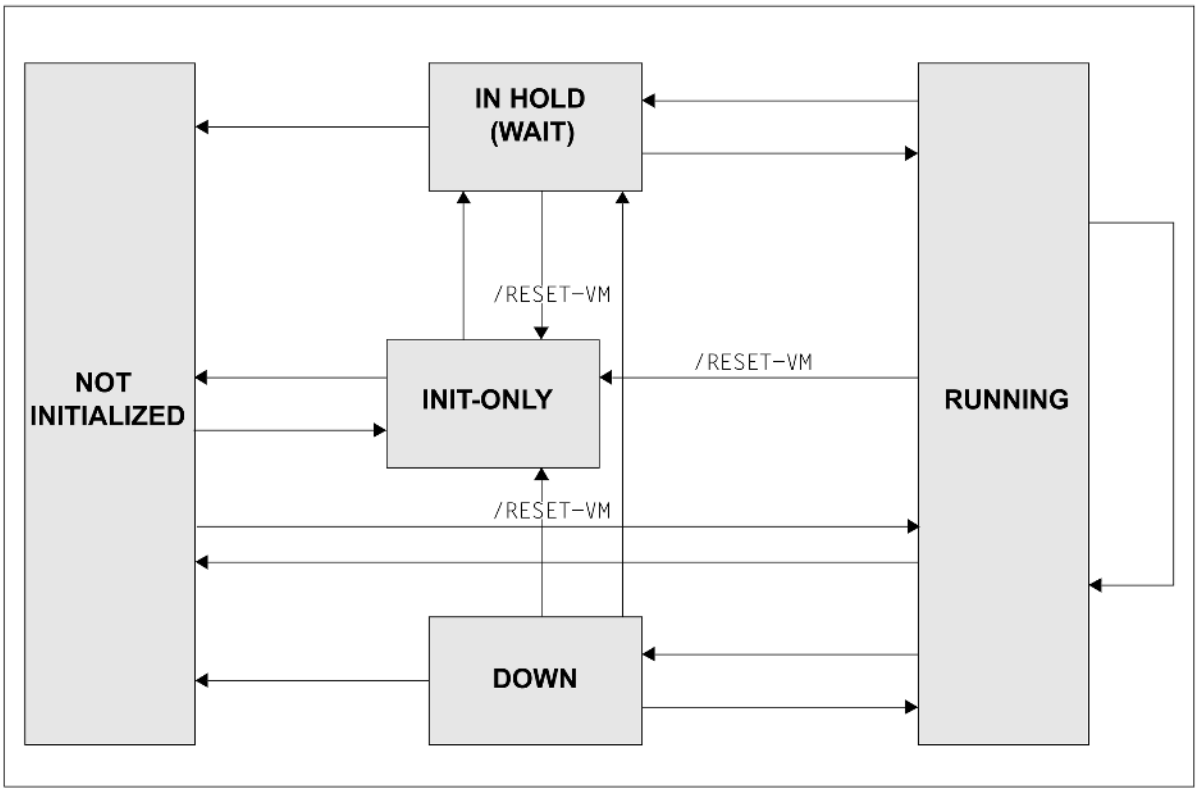


Figure 12: Changing the state of the VM by means of `/RESET-VM`

5.2.16 Shutdown in the guest system and of VM2000

/SHUTDOWN-VM is used to execute a shutdown (without a restart) in the selected BS2000 guest systems:

- The VM administrator uses the VM-ID=*CURRENT operand to shut down the guest system on his/her VM.
- The VM2000 administrator uses the VM-ID=<vm-id> operand to shut down an arbitrary 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, 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 $TSOS.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=*NO/*END(RESTART=*NO). The privilege OPERATING is required for the execution of /SHUTDOWN (see "Privileges of the user task in the monitor system").

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 shutdown (/SHUTDOWN MODE=*NO/*END(RESTART=*NO)) is specified in minutes.

Shutdown for a guest system by the SE manager

Following confirmation and depending on the BS2000 system parameter SYSPROC, the SE manager either directly initiates the shutdown of BS2000 or starts the $TSOS.SYSENT.SHUTDOWN enter job defined for this purpose. A shutdown of the monitor system is only possible after all guest systems have been shut down.

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 with the exception of the monitor system, the shutdown is initiated as described above.

   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.

> The maximum wait time selected should be greater than the longest SHUTCTL wait time in the BS2000 guest systems, see system parameter SHUTCTL in the “Introduction to System Administration” [2]. The guest systems can consequently be shut down before the monitor system.

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)
- `/ACTIVATE-VM-DEFINITION`
- `/MIGRATE-VM`
- On SU /390 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. When the monitor system is shut down, VM2000 performs final actions, see "Coordinated shutdown of VM2000".

   390 On SU /390, the shutdown of the monitor system also terminates operation of VM2000 and the Server Unit.

   x86 The SU x86 continues to operate even after the monitor system has shut down. It can be shut down using the SE Manager.

An example of a coordinated shutdown is provided in the section "Coordinated shutdown of VM2000".

A new IPL can then take place on all Server Units.

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 “Introduction to System Administration” [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.

  x86 On SU x86, VM2000 **cannot** prevent the guest system from being started or restarted via the SE 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.

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.

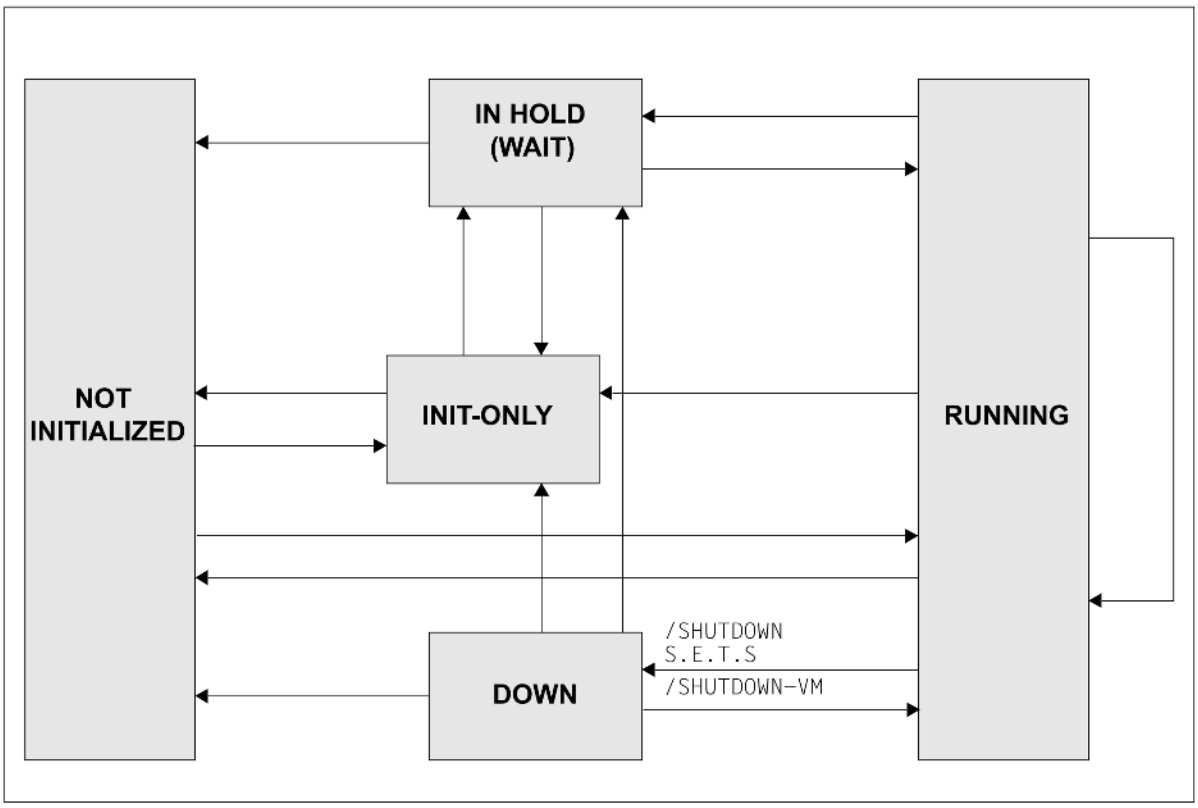


Figure 13: Changing the state of a VM by means of shutdown

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.

x86 On SU x86, the main memory is released asynchronously and this can take several seconds. Only then is the released memory available for other or new VMs.

i The VM definition of a non-persistent VM is deleted. The VM definition of a persistent VM is retained. Further information on VM definitions is provided in the section "Working with VM definitions".

/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 a VM on which the guest system is still active (CHECK-VM-STATE=*YES).
If a VM in the IN HOLD (WAIT) or RUNNING state is to 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.

A persistent VM is assigned the DEFINED_ONLY state in the SE Manager.

Executing /DELETE-VM implicitly executes /HOLD-VM first.

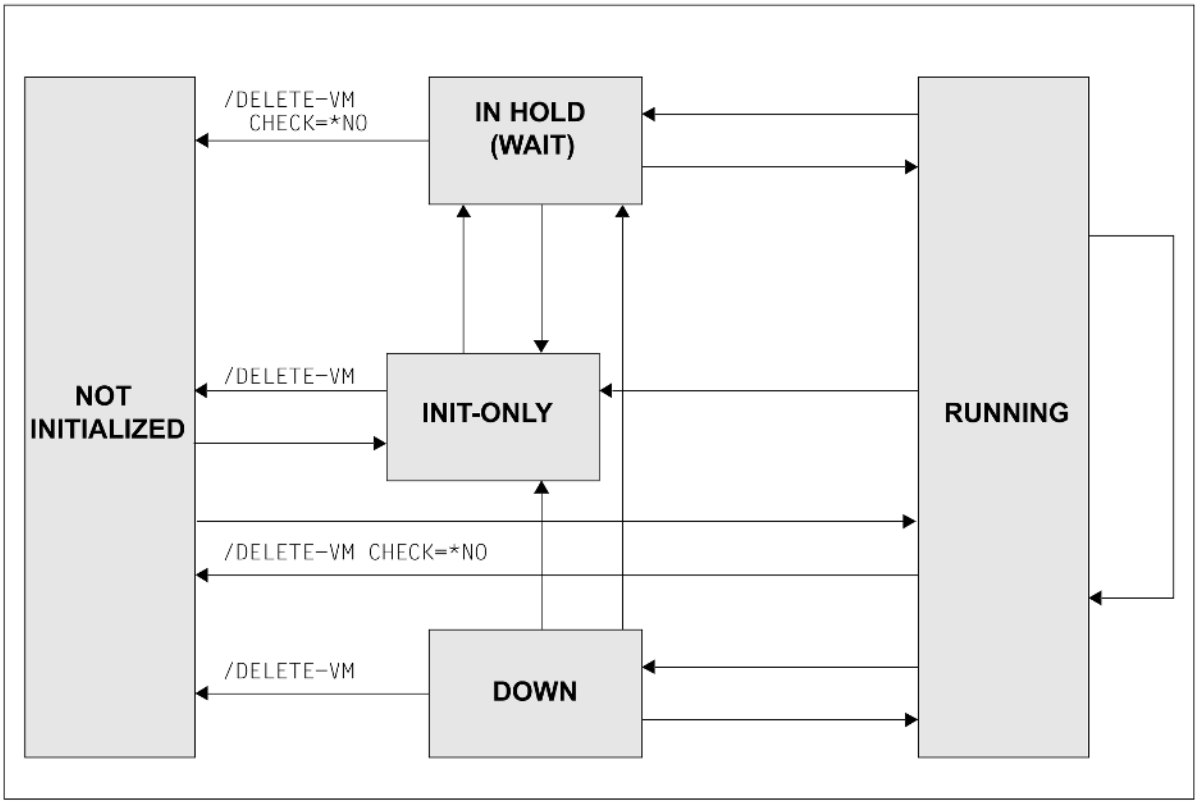


Figure 14: Changing the state of the VM by means of `/DELETE-VM`

5.3 VM Migration

On a Server Unit /390 or x86, a virtual machine can be migrated during guest system operation (VM state RUNNING) from the local Server Unit (source SU) to another Server Unit (target SU) of the same type, using the `/MIGRATE-VM` VM2000 command, while retaining its resources.

This is called **VM Migration** in this manual. Synonymously, the term **Live Migration** is used in BS2000.

During the VM Migration, the VM is only stopped for a short time. User operation, network transfers and transactions on the guest system continue normally on the target SU.

> VMs in INIT-ONLY or DOWN state can be migrated as well.
>
> The persistent VM definition of a VM that has not been initialized can be migrated using `/MIGRATE-VM-DEFINITION`, see section "Relocation of the VM definition during VM Migration".

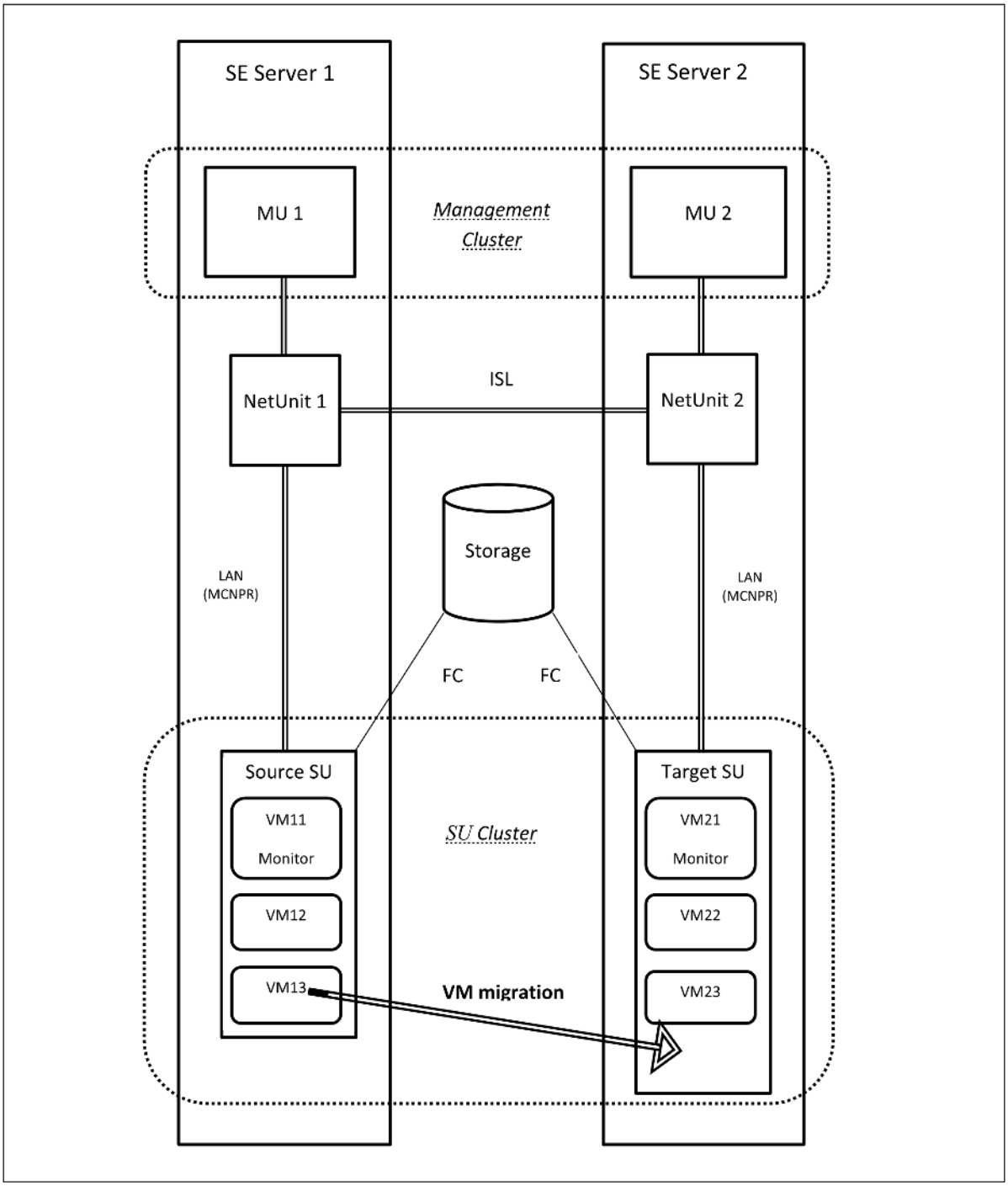


Figure 15: Schematic overview on VM Migration

5.3.1 Prerequisites for the VM Migration

A VM Migration requires the Server Units involved to fulfill some prerequisites. All hard- and software prerequisites are checked by Customer Service and, if necessary, corrected during installation of the SU cluster.

> ¡ You will find information on hard- and software dependencies in the release notice for VM2000. There, you can also find further information on prerequisites and recommendations for VM Migration.

For a successful VM Migration, the following prerequisites have to be fulfilled in the current operation.

General prerequisite

Source and target SU are in an activated SU cluster.

Prerequisites for SU /390

- The main memory size of the VM2000 hypervisor is large enough on both the source and target SU, see HYPERVISOR-SIZE parameter in the section "Setting the VM2000 parameters in the parameter service".

  > ¡ If the hypervisor size is insufficient, the VM Migration and, potentially, other hypervisor activities cannot be executed.

- The main memory size of the monitor VM is at least 512 Mbytes on both the source and target SU, see section "Configuration via the VM2000 configuration file (SU /390)". If the monitor system is working productively, the main memory size must be appropriately higher.
- The monitor system on both source and target SU has a suitable CPU capacity (multiprocessor VM, sufficient CPU quota, unrestricted CPU utilization).

  > ¡ Insufficient main memory size or CPU capacity on the monitor VM prevent the VM Migration from being executed or from being executed efficiently.

- Source and target SU should have comparable I/O configurations. It is best to use an identical, if necessary “over configured”, IOCF, meaning a shared hardware generation for the I/O configurations of both Server Units, which may include connections for BS2000 devices that are not (yet) physically present on one of the SUs. The device configuration must at least be suitable for those devices that are to be allocated to a VM to be migrated.

VM2000 assumes that these prerequisites are met. They are **not** checked in the `/CHECK-VM-MIGRATION` and `/MIGRATE-VM` commands. These only check if the SU cluster is currently active and whether the I/O configurations of the devices allocated to the VM to be migrated are suitable.

Prerequisites for the source SU

- The monitor VM cannot be migrated.
- No other VM Migrations from this source SU and no VM2000 shutdown are currently being executed.
- Additional prerequisite if the VM to be migrated is in DOWN state:
  - The guest system was not terminated abnormally (SETS).

- Additional prerequisites if the VM to be migrated is in RUNNING state:
  - The guest system is ready to operate (“system ready”).
  - The VM has not been stopped with `/HOLD-VM` (SU /390).
  - The main memory of the VM is not currently being reconfigured.
  - There is currently no dynamic I/O configuration modification going on (SU /390).
  - The guest system supports the “Live Migration”.

VM2000 checks these prerequisites in the `/CHECK-VM-MIGRATION` and `/MIGRATE-VM` commands.

Prerequisites for the target SU

If the prerequisites for the source SU are met, VM2000 checks the following prerequisites for the target SU in the `/CHECK-VM-MIGRATION` and `/MIGRATE-VM` commands.

- There are currently no VM2000 shutdown and no VM Migration to this target SU being executed.
- The VM index (if explicitly stated while the VM was being created) and VM name are available, e.g. they have not been assigned to a different VM.
- The maximum number of VMs and VM definitions (SU x86) has not been reached.
- There is enough main memory for the VM to be migrated.
- If the VM to be migrated is assigned to a VM group (SU /390): the VM group exists.
- The CPU pool of the VM to be migrated or of the respective VM group (SU /390) exists and has at least one attached CPU.
- The assignment sets assigned to the VM to be migrated exist.
- The devices assigned to the VM to be migrated have the same device type on both SUs and can be assigned in the same usage mode (`EXCLUSIVE`/`SHARED`).
- Additional prerequisites if the VM to be migrated is in RUNNING state:
  - There is currently no dynamic I/O configuration modification going on (SU /390).
  - The devices of the VM to be migrated have a compatible path configuration (SU /390).
  - The attached disks of the VM to be migrated are ready to operate.
  - The disk devices of the VM to be migrated should have been generated in the same disk storage system, using the same device number (LUN). Otherwise, the warning `VMS2472` is issued.

5.3.2 Phases of a VM Migration

Execution of a VM Migration requires several individual steps on both the source and target SU. Depending on the size and load profile of the VM to be migrated, the necessary time can vary and can therefore not be calculated in advance. Because of this, the VM Migration with the `/MIGRATE-VM` command is split into two phases:

- **Synchronous** checking of the aforementioned prerequisites on both the source and target SU. If the prerequisites are not met, the caller is informed.

  During the synchronous checking, the system identifies which devices of the VM have to be implicitly detached during the asynchronous execution of the VM Migration, see "Device management during a VM Migration". For every device identified device, the warning `VMS2471` is issued.

  > The prerequisites on the source and target SU can also be checked in advance, using the `/CHECK-VM-MIGRATION` command.

- If all prerequisites for the VM Migration are met or if only warnings have been issued, the **asynchronous** execution of the VM Migration is initiated. Control is given back to the caller.

VM status during the migration

On the source SU, the current status of the VM is displayed in the VM2000 information commands, followed by the `(MIGR)` display, e.g. `RUNNING(MIGR)`. On the target SU, the VM is displayed with the `SETUP(MIGR)` state. See "SHOW-VM-ATTRIBUTES (Output attributes of a VM)".

VM Migration on SU /390

On SU /390, VM2000 executes the VM Migration on the source and target SU. Messages regarding the VM Migration are output to the console of the monitor system and to the VM2000 administrators of the source and target SU.

On the source SU, the `VMS2450` message indicates the start of the VM Migration.
The `VMS2452` message indicates that the VM has been stopped.
The `VMS4024` message indicates that the VM has been deleted.
The `VMS2451` message indicates that the VM Migration has been completed successfully.

On the target SU, the `VMS2430` message indicates the start of the VM Migration.
The `VMS3050` message indicates that the VM has been created.
The `VMS2432` message indicates that the VM has been continued.
The `VMS2431` message indicates that the VM Migration has been completed successfully.

In case of warnings or errors, informative messages are output on the source and target SU.

VM Migration on SU x86

On SU x86, the VM Migration is executed by Xen/X2000. Messages regarding the VM Migration are output to the console of the monitor system and to the VM2000 administrators of the source and target SU.

On the source SU, the `VMS2450` message indicates the start of the VM Migration.
The `HAL0331` message (only on the guest system console) indicates that the VM has been stopped.
The `VMS6021` message indicates that the VM has been deleted.
The `VMS2451` message indicates that the VM Migration has been completed successfully.

On the target SU, the `VMS6020` message indicates the creation of the VM.
The `HAL0332` message (only on the guest system console) indicates that the VM has been continued.

In case of a VM Migration with warnings or errors, informative messages are output on the source and target SU.

5.3.3 CPU management during a VM Migration

If a different number of CPUs is attached to the CPU pool of the target SU than to the CPU pool of the source SU, the multiprocessor level of the migrated VM is adjusted accordingly on the target SU. This procedure has the same effect as executing the `/ASSIGN-VM-TO-CPU-POOL ATTACHED-VM-CPUS=*ADJUST-NUMBER` VM2000 command. I.e. redundant virtual CPUs of the VM are detached and set to BLOCK status. Additionally available virtual CPUs (in BLOCK state) are attached if necessary. See the operand description in section "ASSIGN-VM-TO-CPU-POOL (Assign a VM to a CPU pool)".

Exception

Virtual CPUs of the VM that have been detached on the source SU by the guest system using `/DETACH-DEVICE` (they are not in BLOCK state), will not be attached again.

> ¡ After a VM Migration, the distribution of CPU capacity (CPU quota and maximum CPU utilization) should be reconsidered and, if necessary, optimized on both the target and source SU.

5.3.4 Device configuration for the VM Migration (SU /390)

It is recommended to use the same I/O configuration (IOCF) on both the source and target SU. If this is **not** the case, the migrated VM will be displayed with its current state in the VM2000 information commands of the target SU, followed by `(DIFF)`, e.g. `RUNNING(DIFF)`, see "SHOW-VM-ATTRIBUTES (Output attributes of a VM)".

The VM will remain in this state until it is migrated back to its original (source) SU or until the persistent VM on the target SU is reactivated using `/DELETE-VM` and `/ACTIVATE-VM-DEFINITION`.

In this state and in the following cases, you will face restrictions on the target SU after the VM Migration:

Device management

- If the migrated VM was originally created or activated on the source SU, VM2000 may have to virtualize the subchannel numbers for the VM on the target SU.
- A device that was not generated on the source SU cannot be allocated to a migrated VM on the target SU, also see paragraph "BS2000-IPL".

Dynamic configuration modification

- A guest system on a migrated VM in `RUNNING(DIFF)` state cannot be part of a dynamic configuration modification in the monitor system on the target SU. For this type of guest system, the operator receives the `NKR0194` message, which has to be answered.

  > **i** If the dynamic configuration modification in the monitor system is executed nevertheless, it will not become active in a guest system of this type and can cause I/O problems in the guest system.

- Also, in a guest system on a migrated VM that is not in `RUNNING(DIFF)` state, some path specific steps of the dynamic configuration modification in the monitor system on the target SU may not be possible to execute. This problem can be solved by starting the guest system on the target SU anew.
- The dynamic configuration modification is one of the prerequisites of the DPAV function of the IORM utility routine. Accordingly, DPAV cannot be executed for a VM in `RUNNING(DIFF)` state.
- If a VM in `RUNNING(DIFF)` state is migrated back to its original (source) SU and the I/O configuration of that SU has changed in the meantime (e.g. because of a dynamic configuration modification), the VM remains in `...(DIFF)` state on the original (source) SU as well.

BS2000-IPL

In case of a BS2000 IPL, the I/O configuration of the active IOCF of the local (target) SU is used in the guest system. However, the subchannel configuration of the migrated VM under VM2000 corresponds to the I/O configuration of the source SU on which the VM was last created or activated.

This can lead to the following:

- Devices of the I/O configuration of the local (target) SU that were not part of the I/O configuration of the original (source) SU, cannot be assigned to the VM in `...(DIFF)` state, even after a BS2000 IPL (see paragraph "Device management").

- When a VM in `...(DIFF)` state is migrated back to its original (source) SU, this state is usually deleted. Nevertheless, the devices that were not part of the I/O configuration of the other SU can still not be used in the guest system. They can now be allocated to the VM (no longer in `...(DIFF)` state) but not attached in the guest system. The guest system does not know the devices after the BS2000 IPL on the other SU. This problem can be solved by starting the guest system anew.

5.3.5 Device management during a VM Migration

Performing a VM Migration changes the reconfiguration states of the following devices of the VM:

- Tape devices
- Alias devices (PAV, SU /390)
- Locally emulated disk devices (SU x86) or disk devices emulated on an MU (SU /390)

During the asynchronous execution of the VM Migration, these devices are implicitly detached on the source SU. During the synchronous checking of the prerequisites, the caller receives a warning for each of these devices.

After a successful VM Migration on SU /390, the alias devices are attached again on the target SU. Tape devices and local disk devices remain detached on the target SU.

If the VM Migration was not successful, all of these devices are attached again on the source SU.

> **i** When alias devices are detached, they are also removed from the VM. When they are attached, they are reassigned to the VM. Depending on the PAV configuration, alias devices on the source SU can be different from those on the target SU.

5.3.6 Relocation of the VM definition during VM Migration

During a VM Migration, the VM definition of the VM is being copied from the source to the target SU and afterwards deleted on the source SU.

If it is not possible to create the VM definition on the target SU, the VM Migration is discontinued.

If the VM definition cannot be deleted on the source SU, a warning is output to the console of the monitor system and to the VM administrators of the source SU. The VM Migration is still executed. VM2000 will delete the VM definition on the source SU automatically later on.

> ¡ The VM definition of a VM that has not been initialized can be migrated using `/MIGRATE-VM-DEFINITION`.

5.3.7 Commands not allowed during a VM Migration

For a VM that is currently being migrated, no actions may be initiated or modifications of the resources performed on either the source or target SU. 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)
  - /ADD-VM-TO-VM-GROUP and /REMOVE-VM-FROM-VM-GROUP (SU /390)
  - /ASSIGN-VM-TO-CPU-POOL
  - /ASSIGN-VM-GROUP-TO-CPU-POOL
    (SU /390, if a VM of the VM group is being migrated)
  - /DELETE-VM (CHECK-VM-STATE=*YES)
  - /EXTEND-VM-MEMORY, /REDUCE-VM-MEMORY
  - /HOLD-VM (SU /390)
  - /MODIFY-VM-ATTRIBUTES
  - /MOVE-VM (SU /390)
  - /RESET-VM
  - /SHUTDOWN-VM
  - /START-VM (CHECK-VM-STATE=*YES),
    on the target SU also /START-VM (CHECK-VM-STATE=*NO)
- The following VM2000 commands are always rejected during VM Migration with the VMS3042 message:
  - /SAVE-VM-CONFIGURATION
  - /SHUTDOWN-VM VM-ID=*VM2000(...)
- The following VM2000 commands are only partially executed during VM 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 (SU /390)
  - /SHUTDOWN-VM VM-ID=*VM-GUESTS
- On SU /390, the dynamic I/O configuration modification with the /START-CONFIGURATION-UPDATE BS2000 command is generally rejected with the message NKR0197.

5.3.8 Monitoring a VM Migration with job variables (MONJV)

When the VM Migration is initialized by the VM2000 administrator with `/MIGRATE-VM`, a monitoring job variable (MONJV) can be 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 while the VM Migration 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 in VM2000, see "Monitoring a VM with job variables (MONJV)".

MONJVs can also be used to monitor a VM and for main memory reconfiguration, see "Monitoring a VM with job variables (MONJV)" and "Reconfiguring main memory". You can find examples for using MONJV in these sections.

Beginning and end of monitoring

The MONJV used to monitor the VM Migration is configured when the VM Migration is initialized with `/MIGRATE-VM` by specifying the operand `MONJV`. The monitoring of the VM Migration begins at this point; the MONJV is supplied with data by VM2000.

If a MONJV already exists with the specified name and with a password, this password can be specified in the operand `JV-PASSWORD` to check for access authorization.

As soon as the VM Migration is regularly or irregularly terminated, the monitoring is stopped as well. Even when the monitoring has finished, the MONJV is retained as a user job variable without MONJV protection.

Protecting the MONJV

See the paragraph with this name in the section "Monitoring a VM with job variables (MONJV)".

Location of the MONJV

See the paragraph with this name in the section "Monitoring a VM with job variables (MONJV)".

Values for the MONJV

A MONJV for a VM Migration 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 during the execution of the VM Migration.

Byte	Meaning / possible values
1-3	Monitoring status (left-aligned, padded with a space): • `$S` : VM Migration initiated • `$R` : VM Migration running • `$T` : VM Migration successfully terminated • `$TW` : VM Migration terminated with warnings • `$AE` : VM Migration aborted due to error • `$AF` : VM Migration rejected
4	Reserved, has the value zero (0)
5-8	Reserved, contains spaces
9-12	Catalog ID of the home pubset of the monitor system (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 the VM (`<name 1..8>`)
79	Reserved, has the value zero (0)
80-81	Index of VM printable (`<name 2..2>`, e.g. `02`)
82-89	Name of the target SU (`<composed_name 1..8>`, possibly shortened)

90-128	Only for $TW and $AE states: supplementary information For $TW: VM could not be deleted For $AE: Aborted after restart of monitor Monitor memory shortage Connection to target SU disturbed Continuation rejected by guest system Continuation by guest system in delay Detach of device <mn> rejected Detach of device <mn> in delay Device <mn> busy Internal error Aborted by VM2000 on target SU Error recovery or reconfig in progress Aborted by VM system VM deleted regardless to its state Too many changing memory pages Hypervisor memory exhausted
129-256	Reserved for users

Displaying the MONJV

See the paragraph with this name in the section "Monitoring a VM with job variables (MONJV)".

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 `MIGR-MONJV`.

Behavior when the system is terminated abnormally in the monitor system

See the paragraph with this name in the section "Monitoring a VM with job variables (MONJV)".

5.3.9 Example for a VM Migration (SU /390)

Execution on the source SU

```
/migrate-vm vm-id=g4ivo,server-unit-name=d021ze01
%  VMS2405 LIVE MIGRATION OF VM (4,G4IVO) IN CURRENT STATE NOT PERMITTED
%  VMS2462 REQUIREMENTS FOR LIVE MIGRATION OF VM (4,G4IVO) TO TARGET SU
'D021ZE01' NOT FULFILLED. /MIGRATE-VM REJECTED ————————————————————————  (1)
/show-vm-resources
              CPU-POOL(P)
VM-ID         VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 1 M4IVR      MONGROUP(G)   1.00   80.00  2(+1)       512  (100)
 2 G4IVQ      *STDPOOL(P)  25.00  100.00  2(+1)      2048  (100)
 3 G4IVP      USRGROUP(G)   2.00  100.00  2(+1)      3072   100
 4 G4IVO      USRGROUP(G)   2.00  100.00  2(+1)      2048  (100)
 5 G4IVJ      *STDPOOL(P)   5.00  100.00  1(+1)       512  ( 50)
VM-ID         STATE           VERSION PER ADMIN PRIV
 1 M4IVR      RUNNING         V21.0B  NO  YES   IO-R,AG,AS
 2 G4IVQ      RUNNING         V21.0B  YES NO    AG,AS
 3 G4IVP      RUNNING         V20.0B  YES NO    AG,AS
 4 G4IVO      RUNNING         V21.0B  YES NO    AG,AS ————————————————————  (2)
 5 G4IVJ      RUNNING         V20.0B  NO  NO
F R E E       VM: 10          LARGEST-DOMAIN(MB):    7664
/migrate-vm vm-id=g4ivo,server-unit-name=d021ze01
VMS2471 DEVICE '9940' WILL BE DETACHED BY LIVE MIGRATION
VMS2471 DEVICE '9941' WILL BE DETACHED BY LIVE MIGRATION
VMS2471 DEVICE '9942' WILL BE DETACHED BY LIVE MIGRATION ————————————————  (3)
VMS2450 LIVE MIGRATION OF VM (4,G4IVO) TO TARGET SU 'D021ZE01' STARTED—  (4)
/show-vm-resources
VMS4009 DEVICE '9940' REMOVED FROM VIRTUAL MACHINE (4,G4IVO) (BY GUEST)
VMS4009 DEVICE '9941' REMOVED FROM VIRTUAL MACHINE (4,G4IVO) (BY GUEST)
VMS4009 DEVICE '9942' REMOVED FROM VIRTUAL MACHINE (4,G4IVO)
        (BY GUEST) ————————————————————————————————————————————————————  (5)
              CPU-POOL(P)
VM-ID         VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 2 G4IVQ      *STDPOOL(P)  25.00  100.00  2(+1)      2048  (100)
 3 G4IVP      USRGROUP(G)   2.00  100.00  2(+1)      3072   100
 4 G4IVO      USRGROUP(G)   2.00  100.00  2(+1)      2048  (100)
 5 G4IVJ      *STDPOOL(P)   5.00  100.00  1(+1)       512  ( 50)
VM-ID         STATE           VERSION PER ADMIN PRIV
 1 M4IVR      RUNNING         V21.0B  NO  YES   IO-R,AG,AS
 2 G4IVQ      RUNNING         V21.0B  YES NO    AG,AS
 3 G4IVP      RUNNING         V20.0B  YES NO    AG,AS
 4 G4IVO      RUNNING(MIGR)   V21.0B  YES NO    AG,AS ————————————————————  (6)
 5 G4IVJ      RUNNING         V20.0B  NO  NO
F R E E       VM: 10          LARGEST-DOMAIN(MB):    7664
/show-vm-definition vm-id=g4ivo
```

```
                          CPU-POOL(P)
VM-ID         ACT PER HA  VM-GROUP(G)  PROC   MEM-SIZE MAX-IO
 - G4IVO      YES YES NO  USRGROUP(G)   2         2048    100 ————————————  (7)
/show-vm-resources
VMS2452 VM (4,G4IVO) SUSPENDED AT LIVE MIGRATION TO
        TARGET SU 'D021ZE01' ———————————————————————————————————————————  (8)
VMS4009 DEVICE 'CL' REMOVED FROM VIRTUAL MACHINE (4,G4IVO)
VMS4009 DEVICE 'CM' REMOVED FROM VIRTUAL MACHINE (4,G4IVO)
...
VMS4009 DEVICE 'FC15' REMOVED FROM VIRTUAL MACHINE (4,G4IVO)
VMS4429 VIRTUAL MACHINE (4,G4IVO) REMOVED FROM VM GROUP 'USRGROUP'
VMS4619 VIRTUAL MACHINE (4,G4IVO) REMOVED FROM CPU POOL 'USRCPOOL'
VMS4024 VIRTUAL MACHINE (4,G4IVO) TERMINATED ————————————————————————————  (9)
VMS2451 LIVE MIGRATION OF VM (4,G4IVO) TO
        TARGET SU 'D021ZE01' COMPLETED ———————————————————————————————— (10)
             CPU-POOL(P)
VM-ID         VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 1 M4IVR      MONGROUP(G)   1.00   80.00  2(+1)       512  (100)
 2 G4IVQ      *STDPOOL(P)  25.00  100.00  2(+1)      2048  (100)
 3 G4IVP      USRGROUP(G)   2.00  100.00  2(+1)      3072   100
 5 G4IVJ      *STDPOOL(P)   5.00  100.00  1(+1)       512  ( 50)
VM-ID         STATE           VERSION PER ADMIN PRIV
 1 M4IVR      RUNNING         V21.0B  NO  YES   IO-R,AG,AS
 2 G4IVQ      RUNNING         V21.0B  YES NO    AG,AS
 3 G4IVP      RUNNING         V20.0B  YES NO    AG,AS
 5 G4IVJ      RUNNING         V20.0B  NO  NO
F R E E       VM: 11          LARGEST-DOMAIN(MB):    9712 ———————————————— (11)
/show-vm-definition vm-id=g4ivo
VMS2300 DEFINITION OF VM 'G4IVO' DOES NOT EXIST ———————————————————————— (12)
```

---

(1) The guest system on the VM is still in the startup phase. A VM Migration is not possible. If an obstacle is already detected on the source SU, no checks will be performed on the target SU. There will also be no warnings as in the following command `MIGRATE-VM.`

(2) The guest system (V21.0B) on the VM is now ready to operate and is to be migrated.

(3) Warnings: Alias devices are automatically detached and removed from the VM during VM Migration.

(4) Synchronous checking of the prerequisites for the VM Migration has been successfully completed. The asynchronous VM Migration is initiated.

(5) Asynchronous messages on the source SU: The alias devices of the VM are automatically detached and removed from the VM.

(6) During the VM Migration, the VM is displayed in `...(MIGR)` state on the monitor system of the source SU.

(7) The persistent VM definition of the VM still exists on the source SU.

(8) The VM has been stopped on the source SU.

(9) The VM has been terminated on the source SU.

(10) The VM Migration has been terminated (asynchronous termination message by `/MIGRATE-VM`).

(11) The VM is no longer displayed in the VM list. The number of free VMs increases because of this.

(12) The persistent VM definition of the VM no longer exists on the source SU.

Execution on the target SU

```
/show-vm-resources ──────────────────────────────────────────────────────── (13)
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 1 M4IVV     *STDPOOL(P)  20.00  100.00  2(+1)       512  (100)
 2 G4IVK     *STDPOOL(P)  10.00   10.00  1(+1)       512  ( 50)
 3 G4IVL     *STDPOOL(P)  45.00  100.00  4(+1)      2048  (100)
 4 G4IVM     USRGROUP(G)   4.00  100.00  4(+1)      4096  (100)
 5 G4IVN     USRGROUP(G)   1.00  100.00  1(+1)       512  (100)
VM-ID        STATE          VERSION PER ADMIN PRIV
 1 M4IVV     RUNNING        V21.0B  NO  YES   IO-R,AS
 2 G4IVK     RUNNING        V20.0B  NO  NO
 3 G4IVL     RUNNING        V20.0B  NO  NO    AG,AS
 4 G4IVM     RUNNING        V21.0B  YES NO    AS
 5 G4IVN     RUNNING        V21.0B  YES NO    AS
F R E E      VM: 10         LARGEST-DOMAIN(MB):    8176
/show-vm-resources
VMS2430 LIVE MIGRATION OF VM ( ,G4IVO) FROM SOURCE SU 'D020ZE01' STARTED
VMS3050 VIRTUAL MACHINE (6,G4IVO) CREATED
VMS4615 VIRTUAL MACHINE (6,G4IVO) ASSIGNED TO CPU POOL 'USRCPOOL'
VMS4425 VIRTUAL MACHINE (6,G4IVO) ASSIGNED TO VM GROUP 'USRGROUP'
VMS4005 DEVICE 'CL' ASSIGNED TO VIRTUAL MACHINE (6,G4IVO)
VMS4005 DEVICE 'CM' ASSIGNED TO VIRTUAL MACHINE (6,G4IVO)
...
VMS4005 DEVICE 'T1' ASSIGNED TO VIRTUAL MACHINE (6,G4IVO) ──────────────── (14)
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 1 M4IVV     *STDPOOL(P)  20.00  100.00  2(+1)       512  (100)
 2 G4IVK     *STDPOOL(P)  10.00   10.00  1(+1)       512  ( 50)
 3 G4IVL     *STDPOOL(P)  45.00  100.00  4(+1)      2048  (100)
 4 G4IVM     USRGROUP(G)   4.00  100.00  4(+1)      4096  (100)
 5 G4IVN     USRGROUP(G)   1.00  100.00  1(+1)       512  (100)
 6 G4IVO     USRGROUP(G)   2.00  100.00  2(+1)      2048  (100)
VM-ID        STATE          VERSION PER ADMIN PRIV
```

```
 1 M4IVV     RUNNING         V21.0B  NO  YES   IO-R,AS
 2 G4IVK     RUNNING         V20.0B  NO  NO
 3 G4IVL     RUNNING         V20.0B  NO  NO    AG,AS
 4 G4IVM     RUNNING         V21.0B  YES NO    AS
 5 G4IVN     RUNNING         V21.0B  YES NO    AS
 6 G4IVO     SETUP(MIGR)             YES NO    AG,AS ———————————————————— (15)
F R E E      VM:  9          LARGEST-DOMAIN(MB):    6128
/show-vm-resources
VMS2432 VM (6,G4IVO) RESUMED AT LIVE MIGRATION FROM
        SOURCE SU 'D020ZE01' ———————————————————————————————————————— (16)
VMS4621 IMPLICIT DETACH-DEVICE FOR VIRTUAL CPUS ON
        VIRTUAL MACHINE (6,G4IVO) STARTED
VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (6,G4IVO),
        CODE 80FFFFFF, CPU 01
VMS2011 CPU 01 OF VIRTUAL MACHINE (6,G4IVO) STOPPED
VMS2023 CPU 02 OF VIRTUAL MACHINE (6,G4IVO) STARTED ———————————————— (17)
VMS4005 DEVICE '9940' ASSIGNED TO VIRTUAL MACHINE (6,G4IVO)   (BY GUEST)
VMS4005 DEVICE '9941' ASSIGNED TO VIRTUAL MACHINE (6,G4IVO)   (BY GUEST)
VMS4005 DEVICE '9942' ASSIGNED TO VIRTUAL MACHINE (6,G4IVO)
        (BY GUEST) ———————————————————————————————————————————————— (18)
VMS2431 LIVE MIGRATION OF VM (6,G4IVO) FROM SOURCE SU 'D020ZE01'
        COMPLETED ————————————————————————————————————————————————— (19)
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 1 M4IVV     *STDPOOL(P)  20.00  100.00  2(+1)       512  (100)
 2 G4IVK     *STDPOOL(P)  10.00   10.00  1(+1)       512  ( 50)
 3 G4IVL     *STDPOOL(P)  45.00  100.00  4(+1)      2048  (100)
 4 G4IVM     USRGROUP(G)   4.00  100.00  4(+1)      4096  (100)
 5 G4IVN     USRGROUP(G)   1.00  100.00  1(+1)       512  (100)
 6 G4IVO     USRGROUP(G)   2.00  100.00  2(+1)      2048  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 1 M4IVV     RUNNING         V21.0B  NO  YES   IO-R,AS
 2 G4IVK     RUNNING         V20.0B  NO  NO
 3 G4IVL     RUNNING         V20.0B  NO  NO    AG,AS
 4 G4IVM     RUNNING         V21.0B  YES NO    AS
 5 G4IVN     RUNNING         V21.0B  YES NO    AS
 6 G4IVO     RUNNING(DIFF)   V21.0B  YES NO    AG,AS ———————————————— (20)
F R E E      VM:  9          LARGEST-DOMAIN(MB):    6128
/show-vm-definition vm-id=g4ivo
                        CPU-POOL(P)
VM-ID        ACT PER HA  VM-GROUP(G)  PROC   MEM-SIZE MAX-IO
 - G4IVO     YES YES NO  USRGROUP(G)   2         2048    100 ———————— (21)
```

---

(13) Prior to the VM Migration, 5 VMs are running on the target SU. 10 further VMs can be created.

(14) Start of the VM Migration and additional asynchronous messages during creation of the VM.

(15) During the VM Migration, the VM is displayed in `SETUP(MIGR)` state on the monitor system of the target SU. The number of free VMs decreases because of the VM Migration.

(16) The VM resumes operation on the target SU.

(17) Because the relevant CPU pool on the target SU has only one real CPU, a CPU reconfiguration is performed in the guest system. The normal CPU `01` is detached. The spare CPU `02` is started because the VM now only runs in the MONO multiprocessor level.

(18) The alias devices are automatically attached and implicitly assigned to the VM.

(19) The VM Migration is complete.

(20) The VM now runs on the target SU. The I/O configurations of source and target SU are different from each other. The VM continues to run with its original (virtualized) I/O configuration.

(21) The persistent VM definition of the VM now exists on the target SU.

Console outputs of the monitor system on the source SU

```
%VM2M-000.141533  %  VMS2450 LIVE MIGRATION OF VM (4,G4IVO) TO
                     TARGET SU 'D021ZE01' STARTED
%XAAM-000.141612  %  VMS2452 VM (4,G4IVO) SUSPENDED AT LIVE MIGRATION TO
                     TARGET SU 'D021ZE01'
%VM2M-000.141613  %  VMS4024 VIRTUAL MACHINE (4,G4IVO) TERMINATED
%VM2M-000.141614  %  VMS2451 LIVE MIGRATION OF VM (4,G4IVO) TO
                     TARGET SU 'D021ZE01' COMPLETED
```

Console outputs of the monitor system on the target SU

```
%VM2M-000.141535  %  VMS2430 LIVE MIGRATION OF VM ( ,G4IVO) FROM
                     SOURCE SU 'D020ZE01' STARTED
%VM2M-000.141535  %  VMS3050 VIRTUAL MACHINE (6,G4IVO) CREATED
%VM2T-000.141612  %  VMS2432 VM (6,G4IVO) RESUMED AT LIVE MIGRATION FROM
                     SOURCE SU 'D020ZE01'
%VM2G-000.141612  %  VMS4621 IMPLICIT DETACH-DEVICE FOR VIRTUAL CPUS ON
                     VIRTUAL MACHINE (6,G4IVO) STARTED
%VM2G-000.141612  %  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR
                     VIRTUAL MACHINE (6,G4IVO), CODE 80FFFFFF, CPU 01
%VM2G-000.141612  %  VMS2011 CPU 01 OF VIRTUAL MACHINE (6,G4IVO) STOPPED
%VM2G-000.141615  %  VMS2023 CPU 02 OF VIRTUAL MACHINE (6,G4IVO) STARTED
%VM2T-000.141617  %  VMS2431 LIVE MIGRATION OF VM (6,G4IVO) FROM
                     SOURCE SU 'D020ZE01' COMPLETED
```

Excerpt from the CONSLOG file of the migrated guest system

```
<R %DRT0-000.141531  %  MCS1048 SYSTEM COMPONENT 'ETMDR' REQUESTS CLUSTER
RECOVERY LOCK
<R %DRT0-000.141531  %  MCS1112 SYSTEM COMPONENT 'ETMDR' HOLDS CLUSTER
RECOVERY LOCK
<* %DRT0-000.141531  %  ETMDR01 DYNAMIC RECONFIGURATION WITH FUNCTION
'MIGRATE' STARTED ———————————————————————————————————— (22)
<* %XACC-000.141531  %  NKR0074 ALIAS DEVICE '9940' UNLINKED FROM BASE DEVICE
'9930' ——————————————————————————————————————————————— (23)
<G % MSG-000.141531  %  NKR0045 *** 'DEVICE     =9940' DETACHED ***
<* %XACD-000.141532  %  NKR0074 ALIAS DEVICE '9941' UNLINKED FROM BASE DEVICE
'9931'
<G % MSG-000.141532  %  NKR0045 *** 'DEVICE     =9941' DETACHED ***
<* %XACE-000.141533  %  NKR0074 ALIAS DEVICE '9942' UNLINKED FROM BASE DEVICE
'9932'
<G % MSG-000.141533  %  NKR0045 *** 'DEVICE     =9942' DETACHED ***
<* %0JN0-000.141546  %  SVTS005 Service $REWSERV01829172: Normal termination
<W %0JN0-000.141546  %  EXC0420 /LOGOFF PROCESSED. CPU TIME USED: 0.1526 SEC,
USER ID: SYSHSMS, TASK ID: 000100BB, JOB NAME: REWSERV
<* %0JNK-000.141609  %  SVTS005 Service $REWMONG: Normal termination
<W %0JNK-000.141609  %  EXC0420 /LOGOFF PROCESSED. CPU TIME USED: 0.3724 SEC,
USER ID: SYSWSA, TASK ID: 000100AC, JOB NAME: REWMONG ———————————— (24)
<C %BCAM-000.141616  %  BCA0558 CHECKSUM done implicitly by BCAM for VLAN-
LINE "L2#MCNPR"
<* % MSG-000.141616  %  ETMRK19 CPU 01 DETACHED **********************
****** ———————————————————————————————————————————————— (25)
<G % MSG-000.141616  %  NKR0042 'DEVICE     =9940': ATTACH ACCEPTED ——— (26)
<G % MSG-000.141616  %  NKR0042 'DEVICE     =9941': ATTACH ACCEPTED
<G % MSG-000.141616  %  NKR0042 'DEVICE     =9942': ATTACH ACCEPTED
<C %BCAM-000.141616  %  BCA05EE VLAN-LINE "L2#MCNPR" inactive reason: 09
<R %DRT0-000.141616  %  MCS1049 SYSTEM COMPONENT 'ETMDR' RELEASES CLUSTER
RECOVERY LOCK
<C %BCAM-000.141616  %  BCA0571 IPV6-ADDRESS
FD5E:5E5E:600:0:219:99FF:FE9C:7253 switched to VLAN-LINE "L1#MCNPR"
<C %BCAM-000.141616  %  BCA07B2 ROUTE(S) SWITCHED FROM LINE "L2#MCNPR" TO
LINE "L1#MCNPR"; REASON: "1"
<* %XACF-000.141616  %  NKR0073 ALIAS DEVICE '9940' LINKED TO BASE DEVICE
'9930'
<G % MSG-000.141616  %  NKR0040 'DEVICE     =9940' ATTACHED
<* %XACG-000.141616  %  NKR0073 ALIAS DEVICE '9941' LINKED TO BASE DEVICE
'9931'
<* %DRT0-000.141616  %  ETMDR02 DYNAMIC RECONFIGURATION WITH FUNCTION
'MIGRATE' FINISHED
<C %BCAM-000.141616  %  BCA0554 MULTICAST MODE SWITCHED OFF for VLAN-LINE
"L2#MCNPR"
```

```
<C %BCAM-000.141616  %  BCA0558 CHECKSUM done implicitly by BCAM for VLAN-
LINE  "L2#MANPU"
<C %BCAM-000.141616  %  BCA05EE VLAN-LINE "L2#MANPU" inactive reason: 09
<C %BCAM-000.141616  %  BCA0571 IP-ADDRESS 1.1.67.33 switched to VLAN-LINE
"L1#MANPU"
<C %BCAM-000.141616  %  BCA0554 MULTICAST MODE SWITCHED OFF for VLAN-LINE
"L2#MANPU"
<C %BCAM-000.141616  %  BCA0558 CHECKSUM done implicitly by BCAM for VLAN-
LINE "L1#MCNPR"
<C %BCAM-000.141616  %  BCA05EE VLAN-LINE "L1#MCNPR" inactive reason: 09
<G % MSG-000.141616  %  NKR0040 'DEVICE     =9941' ATTACHED
<* %XACH-000.141616  %  NKR0073 ALIAS DEVICE '9942' LINKED TO BASE DEVICE
'9932'
<G % MSG-000.141616  %  NKR0040 'DEVICE     =9942' ATTACHED
<C %BCAM-000.141616  %  BCA0552 BROADCAST MODE SWITCHED OFF for VLAN-LINE
"L1#MCNPR"
<C %BCAM-000.141616  %  BCA0554 MULTICAST MODE SWITCHED OFF for VLAN-LINE
"L1#MCNPR"
<C %BCAM-000.141616  %  BCA0558 CHECKSUM done implicitly by BCAM for VLAN-
LINE "L1#MANPU"
<C %BCAM-000.141616  %  BCA05EE VLAN-LINE "L1#MANPU" inactive reason: 09
<C %BCAM-000.141616  %  BCA0552 BROADCAST MODE SWITCHED OFF for VLAN-LINE
"L1#MANPU"
<C %BCAM-000.141616  %  BCA0554 MULTICAST MODE SWITCHED OFF for VLAN-LINE
"L1#MANPU"
<* %DRT0-000.141616  %  ETMDR03 LIVE MIGRATION FROM SOURCE CONFIGURATION
'390SU-  710-20' TO TARGET CONFIGURATION '390SU-  710-40' FINISHED. INITIAL
CONFIGURATION WAS '390SU-  710-20' ──────────────────────────────────── (27)
<C %BCAM-000.141616  %  BCA05DC VLAN-DEVICE "D1#SE" active
<C %BCAM-000.141616  %  BCA05DC VLAN-DEVICE "D2#SE" active
...
<* %0JGU-000.141617  %  SVTS001 Service $REWMONG: started ──────────────── (28)
<J %0JN1-000.141617  %  JMS0154 'SYSWSA' LOGGED ON FOR 'SUB'. JOB NAME
'REWMONG'. CALLER 'TSN 0JGU'. TID 000100C2
<* %0JN1-000.141618  %  BLS0519 PROGRAM 'REWMONG' LOADED
<C %BCAM-000.141618  %  BCA07D0 LINE "L#MANLO1": "MAX-LPDU-SIZE" changed to
"1514(*BY-DEVICE)"
<C %BCAM-000.141618  %  BCA0847 CHECKSUM done by DEVICE for LINE "L#MANLO1"
<C %BCAM-000.141618  %  BCA07DC LINE "L#MANLO1" active
<C %BCAM-000.141618  %  BCA0841 BROADCAST activated for LINE "L#MANLO1"
<C %BCAM-000.141618  %  BCA0843 MULTICAST activated for LINE "L#MANLO1"
<C %BCAM-000.141618  %  BCA07D0 LINE "L#MANLO2": "MAX-LPDU-SIZE" changed to
"9014(*BY-DEVICE)"
<C %BCAM-000.141618  %  BCA0847 CHECKSUM done by DEVICE for LINE "L#MANLO2"
<C %BCAM-000.141618  %  BCA07DC LINE "L#MANLO2" active
<C %BCAM-000.141618  %  BCA0841 BROADCAST activated for LINE "L#MANLO2"
<C %BCAM-000.141618  %  BCA0843 MULTICAST activated for LINE "L#MANLO2"
```

```
<J %0JN9-000.141618  %  JMS0154 'SYSWSA' LOGGED ON FOR 'SUB'. JOB NAME
'REWPING'. CALLER 'TSN 0JGU'. TID 000100CA
<J %0JN7-000.141618  %  JMS0154 'SYSWSA' LOGGED ON FOR 'SUB'. JOB NAME
'REWPING'. CALLER 'TSN 0JGU'. TID 000100C8
...
<* %0JN8-000.141621  %  BLS0519 PROGRAM 'REWCURL' LOADED
...
<9 %0JGU-000.141627  %  REW0005 MU 'abgsilver' REWAS READY ON
SU 'su0vm06-se2' ———————————————————————————————————————— (29)
<C %BCAM-000.141628  %  BCA05EC VLAN-LINE "L1#MANPU" active
<C %BCAM-000.141628  %  BCA05EC VLAN-LINE "L1#MCNPR" active
<C %BCAM-000.141628  %  BCA05EC VLAN-LINE "L2#MANPU" active
<C %BCAM-000.141628  %  BCA05EC VLAN-LINE "L2#MCNPR" active ———————————— (30)
```

---

(22) Start of the VM Migration.

(23) The alias devices are automatically detached.

(24) The services of the REWAS subsystem are terminated.

(25) Automatic CPU reconfiguration after the VM Migration.

(26) The alias devices are automatically attached.

(27) The VM Migration is complete.

(28) The services of the REWAS subsystem are restarted.

(29) The REWAS subsystem is logging on anew at a Management Unit.

(30) BCAM has newly initialized and activated the redundant HNC configuration.

BS2000 information commands in the guest system (before VM Migration)

The relevant information is marked in **bold**.

```
/show-system-information server-unit=*current
CONFIGURATION                       = 390SU-  710-20
CPU-ID-LIST :     ADR   0           = 1002900236000000
                  ADR   1           = 1012900236000000
                  ADR   2           = 1022900236000000
HSI-ATT :         TYPE              = IX
                  ASF               = YES
                  OPERATION-MODE    = VM2000
MEMORY-SIZE                         = 2 GB
MINIMAL-MEMORY-SIZE                 = 2 GB
BS2000-ID :       NAME              = xxxBXS
                  VERSION           = V21.0B0000
                  SERVICE-PACK      = SP 23.2
                  CREATED           = <date>,<time>
IOCONF-ID :       NAME              = IOCONF1
                  VERSION           = V21.0A01
                  CREATED           = <date>,<time>
                  FORMAT            = IORSF01
IPL-TIME                            = <date>,<time>
SYSTEM-CONF :     SYSID             = 183
                  HOME-PUBSET       = 4IVO
                  HOST-NAME         = D020ZE04
                  VM-INDEX          = 4
                  VM-NAME           = G4IVO
                  SYSTEM-NAME       = *NONE
                  SYSPAR-BS2-SEL    = *STD
                  LIVE-MIG-COUNT    = 0
VM2000-VERSION                      = V12.0B
VM2000-MONITOR-   BS2000-VERSION    = V21.0B0000
   SYSTEM:        HOST-NAME         = D020ZE01
SYSTEM-TIME-      ZONE              = +01:00
   PARAMETER:     SEASON            = S
                  SEASON-DIFFERENCE = 01:00
                  PREV-CHANGE-DATE  = <date>,<time>
                  NEXT-CHANGE-DATE  = <date>,<time>
                  SYNCHRONIZATION   = SERVER-CONN-EXT-REF
                  EPOCH             = <epoch>
/show-live-migration-history
LM-COUNT     DATE       TIME     CONFIGURATION    VM
       0    <date>     <time>   390SU-  710-20     4
```

BS2000 information commands in the guest system (after VM Migration)

The relevant information is marked in **bold**.

```
/show-system-information server-unit=*current
CONFIGURATION                      = 390SU-  710-40
CURRENT-CONFIGURATION              = 390SU-  710-40
CPU-ID-LIST :    ADR   0           = 3002900136000000
                 ADR   1           = 3012900136000000
                 ADR   2           = 3022900136000000
HSI-ATT :        TYPE              = IX
                 ASF               = YES
                 OPERATION-MODE    = VM2000
MEMORY-SIZE                        = 2 GB
MINIMAL-MEMORY-SIZE                = 2 GB
BS2000-ID :      NAME              = xxxBXS
                 VERSION           = V21.0B0000
                 SERVICE-PACK      = SP 23.2
                 CREATED           = <date>,<time>
IOCONF-ID :      NAME              = IOCONF2
                 VERSION           = V21.0A01
                 CREATED           = <date>,<time>
                 FORMAT            = IORSF01
IPL-TIME                           = <date>,<time>
SYSTEM-CONF :    SYSID             = 183
                 HOME-PUBSET       = 4IVO
                 HOST-NAME         = D020ZE04
                 VM-INDEX          = 6
                 VM-NAME           = G4IVO
                 SYSTEM-NAME       = *NONE
                 SYSPAR-BS2-SEL    = *STD
                 LIVE-MIG-COUNT    = 1
VM2000-VERSION                     = V12.0B
VM2000-MONITOR-  BS2000-VERSION    = V21.0B0000
   SYSTEM:       HOST-NAME         = D021ZE01
SYSTEM-TIME-     ZONE              = +01:00
   PARAMETER:    SEASON            = S
                 SEASON-DIFFERENCE = 01:00
                 PREV-CHANGE-DATE  = <date>,<time>
                 NEXT-CHANGE-DATE  = <date>,<time>
                 SYNCHRONIZATION   = SERVER-CONN-EXT-REF
                 EPOCH             = <epoch>
/show-live-migration-history
LM-COUNT     DATE       TIME     CONFIGURATION    VM
       0    <date>     <time>   390SU-  710-20     4
       1    <date>     <time>   390SU-  710-40     6
```

5.4 Working with VM definitions

Each BS2000 VM on a Server Unit of an SE server has a configuration description, the **VM definition**.

The VM definition is created in the following cases:

- When a VM is initialized, see "Initializing a VM"
- With `/CREATE-VM-DEFINITION`
- With `/RECOVER-VM-DEFINITION` (transfer in the SU cluster)

VM definitions are stored on a system disk.

390 On SU /390, BCAM must be active in the monitor system in order to work with VM definitions.

A VM definition is identified by the VM name. It contains:

- Attributes and resources which are assigned to the VM (initialized VM) or envisaged for the VM.
- Devices which are assigned to the (initialized) VM or envisaged for the VM.
- Status of the VM (for display in the SE Manager).

The following attributes of a VM are **not** contained in the VM definition:

- Location of the VM in the main memory of VM2000 (SU /390, is defined by VM2000).
- MONJVs (are determined when the VM definition is activated).

A **persistent VM definition** is created in the following cases:

- When initializing the VM with `/CREATE-VM`, `PERSISTENT=*YES` operand.
- When modifying the VM attributes with `/MODIFY-VM-ATTRIBUTES`, `PERSISTENT=*YES` operand.
- When creating a VM definition with `/CREATE-VM-DEFINITION`.
- When creating a VM definition using the SE Manager.

The monitor VM and other VMs have a non-persistent VM definition.

A persistent VM definition is also available to a Server Unit after a reboot. With its help a persistent VM is set up again and restarted immediately when the corresponding specifications are provided in the `AUTO-IPL` parameter.
A persistent VM definition is **not** deleted by `/DELETE-VM`.

Requirements (SU /390)

On SU /390, VM2000 requires an active BCAM in the monitor system to work with VM definitions on the system disk.

When BCAM is active again in the monitor system and VM2000 was able to perform a complete data comparison with the system disk, message VMS2330 is issued.
Until this occurs, all VM2000 commands which directly concern the VM definitions (e.g. /ACTIVATE-VM-DEFINITION or /SHOW-VM-DEFINITION), or set the PERSISTENT attribute (PERSISTENT=*YES operand in /CREATE-VM or /MODIFY-VM-ATTRIBUTES) are rejected with message VMS2332.
All other VM2000 commands are executed. The contingent changes in the VM definitions are saved by the VM2000 hypervisor and later (before message VMS2330 is issued) transferred to the system disk.

In the case of planned termination of BCAM (BS2000 command /BCEND) in the monitor system, VM2000 issues message VMS2331. After that VM2000 behaves in the way described above until BCAM is active again (VMS2330).

When BCAM is active in the monitor system but the network connection to the system disk is (temporarily) faulty, VM2000 commands which concern VM definitions may possibly be executed with a warning or rejected with a corresponding message.

After warning VMS2306 or VMS2308, VM2000 attempts periodically to subsequently update the VM definition.

5.4.1 VM definition and previous VM2000 commands

The previous VM2000 commands for operating a VM take into account VM definitions as follows:

When a VM is initialized, a VM definition is created for it. When a persistent VM is initialized, a persistent VM definition is also created for it.

390 If the VM definition cannot be created (SU /390 only), the VM is not initialized (irrespective of the PERSISTENT attribute).

VM2000 commands which modify the configuration (e.g. `/ADD-VM-DEVICES`) or the status (e.g. `/HOLD-VM`) of a VM which has already been installed also result in the VM definition being updated.

390 If the VM definition cannot be updated (SU /390 only), a warning is issued. The update of the VM definition is performed automatically at a later time.

The VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM` can use `/MODIFY-VM-ATTRIBUTES` to assign and remove the PERSISTENT attribute to/from a VM which has already been initialized.

When a VM is terminated with `/DELETE-VM`, the VM definition of a persistent VM is retained. The VM definition of a non-persistent VM is deleted.

390 If the VM definition cannot be deleted (SU /390 only), a warning is issued. The deletion of the VM definition is performed automatically at a later time.

5.4.2 Commands for managing VM definitions

The management of VM definitions is the task of the VM2000 administrator. They have a group of VM2000 commands available for this purpose.

The table below contains the various functions for working with VM definitions using the associated VM2000 commands.

Function	VM2000 command
Creating a VM definition	CREATE-VM-DEFINITION [1]
Activating a VM definition	ACTIVATE-VM-DEFINITION [1] [2]
Modifying a VM definition	MODIFY-VM-DEFINITION [1] [2]
Displaying a VM definition	SHOW-VM-DEFINITION [1]
Migrating a VM definition	MIGRATE-VM-DEFINITION [1] [2]
Restoring a VM definition	RECOVER-VM-DEFINITION [1] [2]
Deleting a VM definition	DELETE-VM-DEFINITION [1] [2]

1 For use by the VM2000 administrator only

2 These commands can only be entered for VM definitions of uninitialized VMs. These commands are rejected for initialized VMs.

These functions are described in more detail in the sections below.

Creating a VM definition

The VM2000 administrator creates a persistent VM definition with `/CREATE-VM-DEFINITION`.

The customary attributes and resources for the VM in VM2000 operation are entered in the VM definition, see section "Initializing a VM". The PERSISTENT attribute is assigned implicitly. The location of the VM in the main memory of VM2000 is defined later when the VM definition is activated.

In addition, devices are entered for the VM which are to be assigned when the VM definition is activated. Specifications for the automatic start of the guest system on the VM can also be entered when the VM definition is activated.

The VM definition can then only be processed with the commands for working with VM definitions. Other VM2000 commands with this VM name as VM ID are rejected.

A persistent VM definition can be activated by the VM2000 administrator with `/ACTIVATE-VM-DEFINITION`.

Activating a VM definition

The VM administrator uses /ACTIVATE-VM-DEFINITION to activate a VM definition, i.e. he/she initializes a persistent VM in accordance with the specifications in the VM definition. The location of the VM in the main memory of VM2000 is defined automatically. The devices entered in the VM definition are explicitly assigned to the VM. If required, the guest system on the VM is started in accordance with the settings for automatic startup.

> ¡ Errors when starting the guest system are not displayed by VM2000. The VM remains in the INIT-ONLY status (as with AUTO-IPL=*NO).

The VM definition may not belong to a VM which has already been initialized.

The VM can be monitored with a monitor job variable.

After the command has been executed successfully, the persistent VM is in the INIT-ONLY status (AUTO-IPL=*NO or AUTO-IPL=*YES, error when starting the guest system) or RUNNING (AUTO-IPL=*YES, no error when starting the guest system), see "Operating a VM".

The VM can then be operated using the VM2000 commands employed before. After this command has been executed successfully, the VM definition is updated accordingly.

Modifying a VM definition

With /MODIFY-VM-DEFINITION the VM2000 administrator can modify the following specifications in a VM definitions:

- The VM's attributes and resources which have been entered (see section "Initializing a VM") with the exception of the VM name
- The set of devices entered for the VM
- The specifications for automatic startup of the guest system on the VM

The VM definition may not belong to a VM which has already been initialized.

Displaying a VM definition

With /SHOW-VM-DEFINITION the VM2000 administrator can display the attributes, resources and devices entered for a VM in the VM definition, as well as information on the automatic startup of the automatic startup of the guest system on the VM.

The VM definition may also belong to a VM which has already been initialized. In the case of such a VM, the VM attributes and the resources and devices assigned to the VM (/SHOW-VM-ATTRIBUTES) normally match the values which are entered in the VM definition (/SHOW-VM-DEFINITION). In exceptional cases when VM definitions cannot be modified (see "Requirements (SU /390)" in section "Working with VM definitions"), values displayed in the SE Manager can differ from those in the previous VM2000 outputs.

Migrating a VM definition

The VM2000 administrator can migrate the VM definition of a yet un-initialized VM from the source SU to a target SU with `/MIGRATE-VM-DEFINITION`.

Prerequisites

- Source and target SU are in the same SU cluster.
- The VM name is available on the target SU, i.e. it has not been assigned to another VM or VM definition.
- The maximum number of VM definitions has not been reached on the target SU.

VM2000 checks these prerequisites in the `/MIGRATE-VM-DEFINITION` command.

Execution

On SU /390, the VM2000 monitor on the source SU, the Management Unit and the VM2000 monitor on the target SU collaborate to migrate the VM definition.

On SU x86, the migration of the VM definition is executed via the VM2000 agent (see "Implementation on SU x86"). Xen/X2000 executes the creation of the VM definition on the target SU and the subsequent deletion of the VM definition on the source SU.

Restoring a VM definition

The VM2000 administrator can restore a VM definition with `/RECOVER-VM-DEFINITION`, i.e. a VM definition is transferred from a remote Server Unit to the local Server Unit.

Prerequisites

- The remote Server Unit has to be in the same SU cluster as the local Server Unit.
- The remote Server Unit is not running.

> **i** The caller needs to check these prerequisites.VM2000 only checks if the remote Server Unit has not recently written periodic entries onto the system disk with the VM definitions. On an SU /390, this situation is also possible if BCAM in the monitor system is not ready to operate. Otherwise, VM2000 assumes that the remote Server Unit is still running.

- The VM name is not in use on the local Server Unit.
- The maximum number of VM definitions has not been reached on the local Server Unit.

Execution

The VM definition is read by VM2000 from the remote Server Unit, adapted to the current conditions and saved on the local Server Unit.
After that, the VM definition is deleted on the remote Server Unit. If the VM definition cannot be deleted (e.g. because the remote Server Unit has recently written periodic entries onto the system disk with the VM definitions), it is deleted from the local Server Unit to which it was previously saved.

Deleting a VM definition

With `/DELETE-VM-DEFINITION` the VM2000 administrator can delete a VM definition.

The VM definition may not belong to a VM which has already been initialized.

5.4.3 Examples of working with VM definitions

Asynchronous messages are marked with %.

Example 1: Converting an existing VM to a persistent VM

In the example below (SU /390), a VM is subsequently assigned the PERSISTENT attribute by the VM2000 administrator while the guest system is running.

```
/show-vm-resources
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 1 M4IVR     MONGROUP(G)   1.00   80.00  2(+1)       512  (100)
 2 G4IVQ     *STDPOOL(P)  26.00  100.00  4(+1)      2048  (100)
 3 G4IVP     USRGROUP(G)   4.00  100.00  2(+1)      4096  (100)
 4 G4IVO     USRGROUP(G)   1.00  100.00  1(+1)       512  (100)
 5 G4IVJ     *STDPOOL(P)  70.00  100.00  2(+1)       512  ( 50)
VM-ID        STATE           VERSION PER ADMIN PRIV ————————————————————  (1)
 1 M4IVR     RUNNING         V21.0B  NO  YES   IO-R,AG,AS
 2 G4IVQ     RUNNING         V21.0B  NO  NO    AG,AS
 3 G4IVP     RUNNING         V20.0B  NO  NO    AS
 4 G4IVO     RUNNING         V21.0B  NO  NO    AG,AS
 5 G4IVJ     DOWN            V20.0B  NO  NO
F R E E      VM: 10          LARGEST-DOMAIN(MB):    8176
/start-vm ipl-unit=fc17,main-console=cc,vm-id=g4ivj ————————————————————  (2)
%  VMS2023 CPU 00 OF VIRTUAL MACHINE (5,G4IVJ) STARTED
%  VMS2023 CPU 01 OF VIRTUAL MACHINE (5,G4IVJ) STARTED
%  VMS2023 CPU 02 OF VIRTUAL MACHINE (5,G4IVJ) STARTED
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (5,G4IVJ),
           CODE 80FFFFFF, CPU 02
%  VMS2011 CPU 02 OF VIRTUAL MACHINE (5,G4IVJ) STOPPED
% VMS2050 GUEST SYSTEM ON VM (5,G4IVJ) READY
/show-vm-attributes vm-id=g4ivj,information=*all ————————————————————  (3)
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 5 G4IVJ     *STDPOOL(P)  70.00  100.00  2(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 5 G4IVJ     RUNNING         V20.0B  NO  NO    AS
VM-ID        CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
 5 G4IVJ     *STDPOOL            70.00/ 42.16 100.00/100.00  2(+1)
VM-ID        SCHED ACT-I  VIRT CPUS
 5 G4IVJ      TS    NO    00(RUN),01(RUN),02(OFF)
VM-ID        VM-MONJV
 5 G4IVJ     NO VM-MONJV ASSIGNED
VM-ID        UNITS
```

```
 5 G4IVJ     ASSIGNED BY VM-ADMIN
             EXCL:   CC(VC),CC48,CC49,CC88,CC89,CD48,CD49,CD88,
                            CD89,FC17,FC18,FC19
             SHARED: 990F
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 5 G4IVJ     00001B10        512        512
VM-ID        PRIV    DETAILS
 5 G4IVJ     AS
/show-vm-definition vm-name=g4ivj,information=*all ──────────────────────  (4)
                         CPU-POOL(P)
VM-ID        ACT PER HA  VM-GROUP(G)  PROC   MEM-SIZE MAX-IO
 5 G4IVJ     YES NO  NO  *STDPOOL(P)   2          512    100
                 CPU-POOL(P)
VM-ID        ACT VM-GROUP(G) CPU-Q  MAX-U PROC  ACT-I
 5 G4IVJ     YES *STDPOOL(P) 70.00 100.00  2    NO
VM-ID        ACT MEM-SIZE MIN-MEM-S MAX-MEM-S
 5 G4IVJ     YES      512       512
VM-ID        ACT PRIV    DETAILS
 5 G4IVJ     YES AS
VM-ID        ACT TYPE EFF IPL-U IPL-C MODE   SYSNAME
 5 G4IVJ     YES AUTO NO
                 LAST YES FC17  CC    FAST
VM-ID        ACT UNITS
 5 G4IVJ     YES ASSIGNED BY VM-ADMIN
                 EXCL:   CC(VC),CC48,CC49,CC88,CC89,CD48,CD49,
                         CD88,CD89,FC17(PR),FC18(PD),FC19(PD) ──────────  (5)
                 SHARED: 990F
/modify-vm-attributes vm-id=g4ivj,
       persistent=*yes(auto-ipl=*yes(ipl-unit=fc17,ipl-console=cn)) ──  (6)
/add-vm-devices units=(cn,co),vm-identification=g4ivj ──────────────────  (7)
VMS4005 DEVICE 'CN' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CO' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
/show-vm-attributes vm-identification=g4ivj ────────────────────────────  (8)
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 5 G4IVJ     *STDPOOL(P)  70.00  100.00  2(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 5 G4IVJ     RUNNING         V20.0B  YES NO    AS
/show-vm-definition vm-name=g4ivj ──────────────────────────────────────  (9)
                         CPU-POOL(P)
VM-ID        ACT PER HA  VM-GROUP(G)  PROC   MEM-SIZE MAX-IO
 5 G4IVJ     YES YES NO  *STDPOOL(P)   2          512    100
/show-vm-definition vm-name=g4ivj,information=*ipl ─────────────────────  (10)
VM-ID        ACT TYPE EFF IPL-U IPL-C MODE   SYSNAME
 5 G4IVJ     YES AUTO YES FC17  CN    AUTO
                 LAST YES FC17  CC    FAST
/show-vm-definition vm-name=g4ivj,information=*units ──────────────────  (11)
```

```
VM-ID        ACT UNITS
 5 G4IVJ     YES ASSIGNED BY VM-ADMIN
                 EXCL:   CC(VC),CN,CO,CC48,CC49,CC88,CC89,CD48,
                         CD49,CD88,CD89,FC17(PR),FC18(PD),FC19(PD)
                 SHARED: 990F
/shutdown-vm vm-identification=g4ivj,action=*shutdown(quiet-time=0) ——— (12)
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (5,G4IVJ) INITIATED
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (5,G4IVJ),
           CODE 80FFFFFF, CPU 00
%  VMS2033 'EXC0557 SHUTDOWN PROCESSING COMPLETED ' FROM VM (5,G4IVJ) VIA SVP
%  VMS2051 GUEST SYSTEM ON VM (5,G4IVJ) DOWN; REASON: SHUTDOWN
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (5,G4IVJ),
           CODE 80FFFFFF, CPU 01
/delete-vm vm-identification=g4ivj ————————————————————————————— (13)
VMS4009 DEVICE 'CC' REMOVED FROM VIRTUAL MACHINE (5,G4IVJ)
VMS4009 DEVICE 'CN' REMOVED FROM VIRTUAL MACHINE (5,G4IVJ)
VMS4009 DEVICE 'CO' REMOVED FROM VIRTUAL MACHINE (5,G4IVJ)
...
VMS4619 VIRTUAL MACHINE (5,G4IVJ) REMOVED FROM CPU POOL '*STDPOOL'
VMS4024 VIRTUAL MACHINE (5,G4IVJ) TERMINATED
/show-vm-definition vm-name=g4ivj ————————————————————————————— (14)
                         CPU-POOL(P)
VM-ID        ACT PER HA  VM-GROUP(G)  PROC   MEM-SIZE MAX-IO
 5 G4IVJ     NO  YES NO  *STDPOOL(P)   2          512    100
/modify-vm-definition vm-name=g4ivj,memory-size=1024,
      processor=*extra-and-normal,devices=*remove(units=cc) ———————— (15)
/show-vm-definition vm-name=g4ivj,information=*memory ————————————— (16)
VM-ID        ACT MEM-SIZE MIN-MEM-S MAX-MEM-S
 5 G4IVJ     NO      1024       512
/show-vm-definition vm-name=g4ivj,information=*cpu
                 CPU-POOL(P)
VM-ID        ACT VM-GROUP(G) CPU-Q  MAX-U PROC  ACT-I
 5 G4IVJ     NO  *STDPOOL(P) 70.00 100.00  4(E) NO
/show-vm-definition vm-name=g4ivj,information=*units
VM-ID        ACT UNITS
 5 G4IVJ     NO  ASSIGNED BY VM-ADMIN
                 EXCL:   CN,CO,CC48,CC49,CC88,CC89,CD48,CD49,
                         CD88,CD89,FC17(PR) ———————————————————————— (17)
                 SHARED: 990F
/activate-vm-definition vm-name=g4ivj ————————————————————————————— (18)
VMS3050 VIRTUAL MACHINE (5,G4IVJ) CREATED
VMS4615 VIRTUAL MACHINE (5,G4IVJ) ASSIGNED TO CPU POOL '*STDPOOL'
VMS4005 DEVICE 'CN' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CO' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CC48' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CC49' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CC88' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
```

```
VMS4005 DEVICE 'CC89' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CD48' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CD49' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CD88' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'CD89' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'FC17' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'FC18' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE 'FC19' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ)
VMS4005 DEVICE '990F' ASSIGNED TO VIRTUAL MACHINE (5,G4IVJ) (TYPE=SD)
%  VMS2023 CPU 00 OF VIRTUAL MACHINE (5,G4IVJ) STARTED
%  VMS2023 CPU 01 OF VIRTUAL MACHINE (5,G4IVJ) STARTED
%  VMS2023 CPU 04 OF VIRTUAL MACHINE (5,G4IVJ) STARTED
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (5,G4IVJ),
           CODE 80FFFFFF, CPU 04
%  VMS2011 CPU 04 OF VIRTUAL MACHINE (5,G4IVJ) STOPPED
%  VMS2050 GUEST SYSTEM ON VM (5,G4IVJ) READY
/show-vm-attributes vm-identification=g4ivj ──────────────────────────── (19)
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 5 G4IVJ     *STDPOOL(P)  70.00  100.00  4(+1)      1024  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 5 G4IVJ     RUNNING         V20.0B  YES NO    AS
/show-vm-definition vm-name=g4ivj,information=*ipl ──────────────────── (20)
VM-ID        ACT TYPE EFF IPL-U IPL-C MODE   SYSNAME
 5 G4IVJ     YES AUTO YES FC17  CN    AUTO
                 LAST YES FC17  CN    AUTO
```

---

(1) In the initial situation five VMs are initialized, all of which have a non-persistent VM definition.

(2) VM5 is switched on and started up on its virtual console.

(3) The attributes of the VM2 are output.

(4) The attributes of the VM are also reflected in its (non-persistent) VM definition.

(5) In this case the disks of the home pubset contain the additional indicators `PR` for the pubres device and `PD` for the other pubres disks because they have been assigned to the VM with `/ADD-VM-DEVICES UNITS=*PUBSET-DEVICES(...)`.

(6) The VM is assigned the PERSISTENT attribute, i.e. its VM definition is labeled as persistent. The guest system on the VM is to be started automatically when the VM definition is activated. However, the startup is not to take place from the virtual console, but from a (new) KVP console with the device pair (`CN,CO`).

(7) The (new) KVP console is assigned to the VM. This can also take place after the `/MODIFY-VM-ATTRIBUTES` command.

(8) The VM now has the PERSISTENT attribute.

(9) The VM definition is now active (VM is initialized) and persistent.

(10) The IPL parameters for automatic startup of the guest system on the VM are set. The IPL parameters of the last startup of the guest system (LAST) can also be seen.

(11) The KVP console (`CN,CO`) is assigned.

(12) The guest system on the VM is terminated.

(13) The VM is terminated.

(14) The persistent VM definition still exists. However, it is no longer active.

(15) The VM definition is modified:The next time it is activated the VM is assigned a larger main memory and the highest possible multiprocessor level. At the same time the virtual console is removed from the VM (which is no longer required).

(16) The changes are visible in the information commands.

(17) Disks `FC18` and `FC19` of the home pubset are no longer listed because they have been assigned as pubset devices. When the VM definition is activated, the other pubset disks are determined and displayed dynamically via the pubres device, see comment (5).

(18) Activating the VM definition causes the VM to be set up again and automatically restarted.

(19) The VM runs with the predefined attributes.

(20) In the VM definition the parameters of the last IPL have changed.

Example 2: Setting up a persistent VM from a VM definition

In the example below (SU x86), a persistent VM definition is set up and activated.

```
/create-vm-definition vm-name=persvm, ————————————————————————————  (1)
  memory-size=2048,min-memory-size=1024,max-memory-size=8192,
  processor=*maximum,cpu-quota=20,admin-privileges=*maximum,
  devices=(*pubset-devices(pubset=vm24,usage=*exclusive),
           *device-range(from=fc0a,to=fc0c,usage=*shared),x6,x7,y6,y7,z6,z7),
  privileges=*parameters(assign-by-guest=*yes),
  admin-privileges=*maximum,
  auto-ipl=*yes(ipl-unit=9924,ipl-console=z6,mode=*fast,system-name=abgafr04)
VMS2301 DEFINITION OF VM 'PERSVM' CREATED
/show-vm-definition vm-name=persvm,information=*all ————————————————  (2)
                          CPU-POOL(P)
VM-ID         ACT PER HA  VM-GROUP(G)  PROC   MEM-SIZE MAX-IO
 - PERSVM     NO  YES NO  *STDPOOL(P)   8(M)      2048    100
                 CPU-POOL(P)
```

```
VM-ID        ACT VM-GROUP(G) CPU-Q  MAX-U PROC  ACT-I
 - PERSVM    NO  *STDPOOL(P) 20.00 100.00  8(M) NO
VM-ID        ACT MEM-SIZE MIN-MEM-S MAX-MEM-S
 - PERSVM    NO      2048      1024      8192
VM-ID        ACT PRIV    DETAILS
 - PERSVM    NO  ADM     *VM-MAX
                 AG      *STDSET
                 AS
VM-ID        ACT TYPE EFF IPL-U IPL-C MODE   SYSNAME
 - PERSVM    NO  AUTO YES 9924  Z6    FAST   ABGAFR04
VM-ID        ACT UNITS
 - PERSVM    NO  ASSIGNED BY VM-ADMIN
                 EXCL:   X6,X7,Y6,Y7,Z6,Z7,9924(PR)
                 SHARED: FC0A,FC0B,FC0C
/activate-vm-definition vm-name=persvm ──────────────────────────────────  (3)
%  VMS6020 VIRTUAL MACHINE (4,PERSVM) CREATED BY SYSTEM ADMINISTRATION
%  VMS4005 DEVICE '9924' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM)   (BY SYSADM)
%  VMS4005 DEVICE 'FC0A' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM) (TYPE=SD)
             (BY SYSADM)
%  VMS4005 DEVICE 'FC0B' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM) (TYPE=SD)
            (BY SYSADM)
%  VMS4005 DEVICE 'FC0C' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM) (TYPE=SD)
            (BY SYSADM)
%  VMS4005 DEVICE 'Z6' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM)   (BY SYSADM)
%  VMS4005 DEVICE 'Z7' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM)   (BY SYSADM)
%  VMS4005 DEVICE 'X6' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM)   (BY SYSADM)
%  VMS4005 DEVICE 'X7' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM)   (BY SYSADM)
%  VMS4005 DEVICE 'Y7' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM)   (BY SYSADM)
%  VMS4005 DEVICE 'Y6' ASSIGNED TO VIRTUAL MACHINE (4,PERSVM)   (BY SYSADM)
%  VMS2023 CPU 00 OF VIRTUAL MACHINE (4,PERSVM) STARTED
%  VMS6023 GUEST SYSTEM ON VM (4,PERSVM) STARTED BY SYSTEM ADMINISTRATION
%  VMS2023 CPU 01 OF VIRTUAL MACHINE (4,PERSVM) STARTED
%  VMS2023 CPU 02 OF VIRTUAL MACHINE (4,PERSVM) STARTED
%  VMS2023 CPU 03 OF VIRTUAL MACHINE (4,PERSVM) STARTED
%  VMS2023 CPU 04 OF VIRTUAL MACHINE (4,PERSVM) STARTED
%  VMS2023 CPU 05 OF VIRTUAL MACHINE (4,PERSVM) STARTED
%  VMS2023 CPU 06 OF VIRTUAL MACHINE (4,PERSVM) STARTED
%  VMS2023 CPU 07 OF VIRTUAL MACHINE (4,PERSVM) STARTED
% VMS2050 GUEST SYSTEM ON VM (4,PERSVM) READY
/show-vm-attributes vm-identification=persvm,information=*all ──────────  (4)
             CPU-POOL(P)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 4 PERSVM    *STDPOOL(P)  20.00  100.00    8        2048  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 4 PERSVM    RUNNING         V21.0B  YES NO    AG,AS
VM-ID        CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
 4 PERSVM    *STDPOOL            20.00/ 20.00 100.00/100.00    8
```

```
VM-ID        SCHED ACT-I  VIRT CPUS
 4 PERSVM     TS    NO    00(RUN),01(RUN),02(RUN),03(RUN),
                          04(RUN),05(RUN),06(RUN),07(RUN)
VM-ID        VM-MONJV
 4 PERSVM    NO VM-MONJV ASSIGNED
VM-ID        UNITS
 4 PERSVM    ASSIGNED BY VM-ADMIN
             EXCL:   X6,X7,Y6,Y7,Z6,Z7,9924
             SHARED: FC0A,FC0B,FC0C
VM-ID        BS2-MEM-S  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 4 PERSVM         1896      2048       1024       8192
VM-ID        PRIV    DETAILS
 4 PERSVM    ADM     *VM-MAX
             AG      *STDSET
             AS
/show-vm-definition vm-name=persvm,information=*ipl ———————————————————  (5)
VM-ID        ACT TYPE EFF IPL-U IPL-C MODE   SYSNAME
 - PERSVM    YES AUTO YES 9924  Z6    FAST   ABGAFR04
                 LAST YES 9924  Z6    FAST   ABGAFR04
/shutdown-vm vm-identification=persvm,action=*shutdown(quiet-time=0) ——  (6)
VMS2054 SHUTDOWN OF GUEST SYSTEM ON VM (4,PERSVM) INITIATED
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 07
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 05
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 03
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 04
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 01
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 00
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 02
%  VMS2033 'EXC0557 SHUTDOWN PROCESSING COMPLETED' FROM VM (4,PERSVM) VIA SVP
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (4,PERSVM),
           CODE 00000000, CPU 06
%  VMS2051 GUEST SYSTEM ON VM (4,PERSVM) DOWN; REASON: SHUTDOWN
/delete-vm vm-identification=persvm ——————————————————————————————————  (7)
VMS4619 VIRTUAL MACHINE (4,PERSVM) REMOVED FROM CPU POOL '*STDPOOL'
VMS4024 VIRTUAL MACHINE (4,PERSVM) TERMINATED
%  VMS4009 DEVICE 'Z6' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE 'Z7' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE 'X6' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE 'X7' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE 'Y6' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
```

```
%  VMS4009 DEVICE 'Y7' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE '9924' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE 'FC0A' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE 'FC0B' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
%  VMS4009 DEVICE 'FC0C' REMOVED FROM VIRTUAL MACHINE (4,PERSVM) (BY SYSADM)
/delete-vm-definition vm-name=persvm ──────────────────────────────────────  (8)
VMS2302 DEFINITION OF VM 'PERSVM' DELETED
```

---

(1) The persistent VM definition is set up.

(2) The settings in the persistent VM definition are displayed.

(3) The VM definition is activated. The BS2000 guest system is started automatically.

> ¡ On SU x86, this command causes all messages to be output asynchronously. On SU /390 some messages (in particular `VMS3050` and `VMS4005)` are output synchronously when the command is executed, see comment (18) in Example 1.

(4) The VM runs with the predefined attributes and resources.

(5) After the (automatic) startup of the guest system on the VM, the parameters of the last IPL are also entered in the VM definition.

(6) The guest system on VM4 is terminated.

(7) VM4 is terminated.

> ¡ When a persistent VM is terminated, the messages `VMS4009` are output asynchronously. Otherwise the messages are output synchronously during command execution, see comment (13) in Example 1.

(8) The persistent VM definition is deleted.

5.5 Operating VM groups (SU /390)

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.

x86 VM groups are not available on SU x86.

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" and "Scheduling procedure".

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".

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] CREATE-VM-DEFINITION [1] [2] MODIFY-VM-DEFINITION [1] [2]
Modifying the attributes of a VM in a VM group	MODIFY-VM-ATTRIBUTES [3]
Removing a VM from a VM group	DELETE-VM [4] DELETE-VM-DEFINITION [1] [2] 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 only for working with VM definitions

3 Command for the VM2000 administrator and those VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM` (see "Extending and restricting the range of commands/functions")

4 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.

5.5.1 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 depends on the architecture of the Server Unit, see "Comparison of VM2000 on SU /390 and SU x86".

Example
A maximum of 15 VMs are permitted on an SU /390. 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`.

5.5.2 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 (see "Initializing a VM") 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 (see "Initializing a VM"). It can be modified later using `/MODIFY-VM-ATTRIBUTES`.

> ℹ `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used to enter the assignment of a VM to a VM group in its VM definition. Detailed information can be found in the section "Working with VM definitions".

5.5.3 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.

5.5.4 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.

> ¡ `/DELETE-VM-DEFINITION` cancels the entered assignment of the VM to a VM group.

5.5.5 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`.

5.5.6 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.

5.5.7 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`.

5.5.8 Example of operating VM groups

The example below illustrates how the VM2000 administrator operates a VM group with two VMs on an SU /390.

```
/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)       512  (100)
 2 C1SYS1    *STDPOOL(P)  20.00   20.00  2(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 1 MONITOR   RUNNING         V21.0B  NO  YES   AS
 2 C1SYS1    RUNNING         V20.0B  NO  NO    AS
F R E E      VM: 13          LARGEST-DOMAIN(MB):   14832
/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'
/create-vm memory-size=512,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)
/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'
/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)       512  (100)
 2 C1SYS1    *STDPOOL(P)  30.00   30.00  2(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 1 MONITOR   RUNNING         V21.0B  NO  YES   AS
 2 C1SYS1    RUNNING         V20.0B  NO  NO    AS
F R E E      VM: 13          LARGEST-DOMAIN(MB):   14832

-------------------------------------------------------------------------------------------------------
```

(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 Unit. The VM2000 administrator has therefore equipped the customer VM `C1SYS1` with a CPU quota and a maximum CPU utilization of 20. 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".

(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.

(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 Unit 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).

(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 Unit'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.

(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.

5.6 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 VM2000 administrator 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.

> ¡ Authorization code “9” is required for operating the whole VM2000 system 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 "Extending and restricting the range of commands /functions").

See also: paragraph “Logging” in sections "Administration dialogs from privileged user tasks" and "Operating the virtual machines (ADMIN dialog via $VMCONS)".

5.6.1 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 "Extending and restricting the range of commands/functions").

5.6.2 Overview of commands for VM2000 operation

Operation group	Command	(V)	(P)	(R)
Initializing a VM	CREATE-VM	x		(4)
Device management 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 Assignment sets, implicit device assignment and release")	CREATE-VM-ASSIGNMENT-SET	x		
	DELETE-VM-ASSIGNMENT-SET	x		
	SHOW-VM-ASSIGNMENT-SET	x		
Starts 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	MODIFY-VM-ATTRIBUTES	x	(6) (9)	
(real CPUs)	ATTACH-VM-RESOURCES	x		
	DETACH-VM-RESOURCES	x		
	EXTEND-VM-MEMORY	x		
(main memory)	REDUCE-VM-MEMORY	x		
Command processing via VM2000 procedure	CALL-VM-PROCEDURE		(1)	(5) (7)
390 Suspending a VM	HOLD-VM		(1) (3)	(4)
390 Canceling a wait state	RESUME-VM		(1) (3)	(4)
390 Moving a VM within main memory	MOVE-VM	x		(4)
Migrating a VM in the SU cluster	CHECK-VM-MIGRATION	x		(4)
	MIGRATE-VM	x		(4)

Operation group	Command	(V)	(P)	(R)
Sends a message to VM administrators	SEND-VM-MESSAGE			
Modifying parameters for $VMCONS	MODIFY-VMCONS-PARAMETERS	x		
Terminates administration or operation	END-VM-DIALOG		(1)	
Aborting a guest system	RESET-VM		(1)	(4)
Shutdown of a guest system or VM2000	SHUTDOWN-VM		(1) (8)	(4)
Terminating a VM	DELETE-VM		(1) (3)	(4)
Working with VM definitions (see also "Working with VM definitions")	CREATE-VM-DEFINITION	x		
	ACTIVATE-VM-DEFINITION	x		(10)
	MIGRATE-VM-DEFINITION	x		(10)
	RECOVER-VM-DEFINITION	x		(10)
	MODIFY-VM-DEFINITION	x		(10)
	SHOW-VM-DEFINITION	x		
	DELETE-VM-DEFINITION	x		(10)
390 Operating VM groups (see also "Operating VM groups (SU /390)")	CREATE-VM-GROUP	x		
	ADD-VM-TO-VM-GROUP	x		
	REMOVE-VM-FROM-VM-GROUP	x		
	MODIFY-VM-GROUP-ATTRIBUTES	x		
	SHOW-VM-GROUP	x		
	DELETE-VM-GROUP	x		
Managing CPU pools (see also "Managing CPU pools")	CREATE-VM-CPU-POOL	x		
	ASSIGN-VM-GROUP-TO-CPU-POOL	x		390
	ASSIGN-VM-TO-CPU-POOL	x		
	SWITCH-VM-CPU	x		
	SHOW-VM-CPU-POOL	x		
	DELETE-VM-CPU-POOL	x		
Managing the VM2000 configuration	CHECK-VM-DISK-CONFIGURATION	x		
	SAVE-VM-CONFIGURATION	x		
	RESTORE-VM-CONFIGURATION	x		

(V) VM2000 Administrator Command

(P) Privileged functions

(R) Restriction

(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 "Extending and restricting the range of commands/functions").

(7) This command cannot be issued from a user task.

(8) The VM2000 administrator may terminate all guest systems using `VM-ID=*VM-GUESTS`, and the overall VM2000 system using `VM-ID=*VM2000`.

(9) The VM2000 administrator may modify the setting for the PERSISTENT attribute and the entries for the automatic start in the VM definition of an initialized VM.

(10) These commands can only be entered for VM definitions of uninitialized VMs.

5.6.3 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)`).

5.6.4 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 migrated (/MIGRATE-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 via the SE 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.
- The monitor VM and its VM definition cannot be made persistent.

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.

5.6.5 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.

Restrictions for the VM2000 administrator

The command/function range for the VM2000 administrator can be restricted:

- 390 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 command/functional scope for the VM2000 administrator can also be entered in the VM definition with `/CREATE-` or `/MODIFY-VM-DEFINITION`. Detailed information can be found in the section "Working with VM definitions".

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)
Information services	SHOW-VM-DEVICE-STATUS	(2)
	SHOW-VM-RESOURCES	(2)
Resource management	MODIFY-VM-ATTRIBUTES	(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`.

5.6.6 VM2000 functions in the SE Manager

Fundamental VM2000 functions for managing the BS2000 VMs are also available to the SE server administrators via the SE Manager. The table below provides an overview of the functions.

These functions are described in detail in the “Operation and Administration” manual [19]. They are not described in detail in this manual.

VM2000 function	VM2000 command	SE Manager ***Systems*** **> SU <SU name> selected** [1]
Initializing a VM Creating a VM definition	/CREATE-VM /CREATE-VM-DEFINITION	*VM administration* tab, *Create new BS2000 VM* button [2]
Activating a VM definition	/ACTIVATE-VM-DEFINITION	VM <vm-name> selected, *Operation* tab, *Activate BS2000 VM* action
Starting the guest system on the VM	/START-VM DIAGNOSTIC-IPL=*NO	VM <vm-name> selected, *Operation* tab, *BS2000 IPL* [2] action
Starting the guest system on the VM (diagnostic IPL)	/START-VM DIAGNOSTIC-IPL=*YES	VM <vm-name> selected, *Operation* tab, *BS2000 Dump IPL* [2] action
Performing a shutdown in the guest system	/SHUTDOWN-VM VM=<vm-id>	VM <vm-name> selected, *Operation* tab, *BS2000 Shutdown* action
Assigning devices explicitly Entering devices in the VM definition	/ADD-VM-DEVICES /MODIFY-VM-DEFINITION DEVICES=*ADD	VM <vm-name> selected, *Disks/KVP/Tape devices* tabs *Assign disk / KVP / ...* button *All devices* tab *Assign devices* button
Removing devices Removing devices from the VM definition	/REMOVE-VM-DEVICES /MODIFY-VM-DEFINITION DEVICES=*REMOVE	VM <vm-name> selected, *Disks/KVP/Tape devices* tabs *Remove* action icon *All devices* tab *Remove devices* button
Changing usage mode of disk devices Changing usage mode of disk devices in the VM definition	/MODIFY-VM-DEVICE-USAGE /MODIFY-VM-DEFINITION DEVICES=*MODIFY	VM <vm-name> selected, *Disks* tab, *Change* action icon

VM2000 function	VM2000 command	SE Manager *Systems* > SU <SU name> selected [1]
Displaying the VM attributes Displaying the attributes of VM definitions	/SHOW-VM-ATTRIBUTES /SHOW-VM-RESOURCES /SHOW-VM-DEFINITION	Tabs *VM administration*, *VM resources*, *VM options* and VM <vm-name> selected, *Operation* tab
Changing the VM attributes Changing the attributes of VM definitions	/MODIFY-VM-ATTRIBUTES /MODIFY-VM-DEFINITION	*VM resources* tab, *Change* [3] action icon and *VM options* tab, *Change* action icon
Displaying the attributes of the CPU pools	/SHOW-VM-CPU-POOL	*VM resources* tab
Assigning a VM to a CPU pool	/ASSIGN-VM-TO-CPU-POOL ATTACHED-VM-CPUS= *CHECK-NUMBER	*VM resources* tab, *Change* action icon
Migrating a VM Migrating a VM definition	/MIGRATE-VM /MIGRATE-VM-DEFINITION	VM <vm-name> selected, *Operation* tab, *Migrate BS2000 VM* action
Terminating a VM	/DELETE-VM	VM <vm-name> selected, *Operation* tab, *Deactivate BS2000 VM* action (persistent VM) *Deactivate and delete BS2000 VM* action (non-persistent VM)
Deleting a VM definition	/DELETE-VM-DEFINITION	*VM administration* tab *Delete* action icon

1 There is the same navigation path for SU /390 as well as for SU x86.

2 Some parameters are assigned default values.

3 Some attributes cannot be changed.

5.7 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 in 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 Units

- The nonprivileged macro VMGINF provides global VM2000 information (see the “Executive Macros” manual [16]).
- Output of global VM2000 information in all guest systems (see "SHOW-SYSTEM-INFORMATION (Display information on the current system)" and the BS2000 command `/SHOW-SYSTEM-INFORMATION` in the “Commands” manual [13]).
- Output of VM2000 information on devices and pubsets in the guest system as well (see "SHOW-DEVICE-CONFIGURATION (Display system configuration and availability)" and the BS2000 command `/SHOW-DEVICE-CONFIGURATION` in the “Commands” manual [13]).
- In the case of `/SHUTDOWN-VM`, VM2000 and the guest systems concerned interwork to execute shutdown processing (see "Terminating the monitor system and VM2000").
- Dynamic main memory reconfiguration is carried out by VM2000 together with the guest system (see "Reconfiguring main memory").
- 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 "Assignment sets, implicit device assignment and release" and the description of the BS2000 reconfiguration commands mentioned there as of "ATTACH-DEVICE (Attach hardware units)" and in the “Commands” manual [13]).
- 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 section "Explicit device assignment and removal").
- 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 I/O limit of the IORM function IOLVM is set on SU /390 by means of VM2000 commands (see "Use of IORM in VM2000 operation" 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 "Measurement with the openSM2 measurement monitor" and the “openSM2” manual [9]).
- 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 "Automatic restart after SETS in the monitor system").
- Guest systems are included in certain steps of the VM Migration.

For SU /390

- In the case of /MOVE-VM, VM2000 and the guest systems concerned interwork to provide protection in the MSCF network (see "Moving a VM within main memory").
- Using BS2000 command /MODIFY-IO-UNIT you can specify in a guest system a different base or alias device that is to be preferred for I/Os of this guest system (see "Parallel access volume (PAV, SU /390)" and the command description in the “Commands” manual [13]).
- Using BS2000 command /MODIFY-IO-OPTIONS you can specify in a guest system whether base or alias devices of feature "FastDPAV" are to be preferred for I/Os of this guest system (see "Parallel access volume (PAV, SU /390)" and the command description in the “Commands” manual [13]).
- SLEDFILES with the dump object VM2000 can be evaluated with DAMP (see "Analyzing SLEDFILEs with DAMP" 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 "Guest system errors" and the “Introduction to System Administration” 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 Units

- 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 "Operating the virtual machines (ADMIN dialog from user task)" and the “Introduction to System Administration” [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 "Measurement with the openSM2 measurement monitor" and the “openSM2” manual [9]).
- Using BS2000 command /MODIFY-IO-OPTIONS you can specify a new timeout value for the specified devices by the operating in the monitor system centrally for all guest systems (see also the command description in the “Commands” manual [13]).

For SU /390

- (Global) reconfiguration of hardware units and paths in all the active guest systems by the operator in the monitor system (see "Reconfiguring Channels, controllers and paths (SU /390)" and the description of the BS2000 reconfiguration commands mentioned there as of "ATTACH-DEVICE (Attach hardware units)" and in the “Commands” manual [13]).
- Interaction between VM2000 and the operating in the monitor system during dynamic I/O configuration modification of channel devices (see "Changing the configuration of channel devices using BS2000 (SU /390)" and the “Introduction to System Administration” manual [2]).

5.7.1 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

UNIT = **...** / ***CHANNEL**(...) / ***CONTR**OLLER(...) / ***CHANNEL-RANGE**(...) / ***PUB**SET-**DEV**ICES(...) / **...**

. . .

***CHANNEL**(...)

| **CHANNEL-PATH-ID** = list-poss(16): <x-text 2..2>

| ,**SCOPE** = ***OWN-SYSTEM-ONLY** / ***VM2000-GLOBAL**

***CONTR**OLLER(...)

| **CONTR**OLLER**-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**

***PUB**SET-**DEV**ICES(...)

| **PUB**SET = list-poss(255): <cat-id 1..4> / ***BY-PUBRES-DEV**ICE(...)

| *BY-PUBRES-DEVICE(...)

| | **UNIT** = list-poss(255): <alphanum-name 2..2> / <x-text 4..4>

. . .

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 in VM2000 operating.

SCOPE = *OWN-SYSTEM-ONLY
The command is carried out only in the local system.

SCOPE = *VM2000-GLOBAL

If entered in the monitor system (SU /390), the command is executed in all guest systems.
If entered in another guest system or in the monitor system of an SU x86, the command is rejected with message `NKR0178`.

5.7.2 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

UNIT = **...** / ***CHANNEL**(...) / ***CONTR**OLLER(...) / ***CHANNEL-RANGE**(...) / ***PUB**SET-**DEV**ICES(...) / **...**

. . .

***CHANNEL**(...)

| **CHANNEL-PATH-ID** = list-poss(16): <x-text 2..2>

| ,**SCOPE** = ***OWN-SYSTEM-ONLY** / ***VM2000-GLOBAL**

***CONTR**OLLER(...)

| **CONTR**OLLER**-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**

***PUB**SET-**DEV**ICES(...)

| **PUB**SET = list-poss(255): <cat-id 1..4> / ***BY-PUBRES-DEV**ICE(...)

| *__BY-PUBRES-DEV__ICE(...)

| | **UNIT** = list-poss(255): <alphanum-name 2..2> / <x-text 4..4>

. . .

,**FORCE** = ***STD** / ***Y**ES / ***NO**(...) / ***UNCOND**ITIONAL**-OFFLINE**

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 in VM2000 operating.

SCOPE = *OWN-SYSTEM-ONLY
The command is carried out only in the local system.

SCOPE = *VM2000-GLOBAL
If entered in the monitor system (SU /390), the command is executed in all guest systems.
If entered in another guest system or in the monitor system of an SU x86, 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 only for the monitor system on SU /390.
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.

5.7.3 INCLUDE-/REMOVE-DEVICE-CONNECTION (Add/Remove logical connections)

x86 Global reconfiguration of hardware units is not available on SU x86. 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**-DEV**ICE**-CONN**ECTION / **REM**OVE**-DEV**ICE**-CONN**ECTION
FROM = ***CHANNEL**(...) / ***CONTR**OLLER(...) / 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 in VM2000 operating.

SCOPE = *OWN-SYSTEM-ONLY
The command is carried out only in the local system.

SCOPE = *VM2000-GLOBAL
If entered in the monitor system (SU /390), the command is executed in all guest systems. If entered in another guest system or in the monitor system of an SU x86, the command is rejected with message NKR0178.

5.7.4 MODIFY-IO-UNIT (Modify configuration properties of an I/O unit)

x86 PAV devices are not available on SU x86, 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 SU x86.

This command allows systems support to modify the configuration properties of controllers or devices dynamically.

Format (VM2000-specific operands)

MODIFY-IO-UNIT

UNIT = ***CONTR**OLLER(...) / ***DEV**ICE(...)

. . .

***DEV**ICE(...)

| **NAME** = <alphanum-name 2..2> / <x-text 4..4> / ***RANGE**(...)

| ,**STATE** = ***UNCHA**NGED / ***PREFERRED-PATH**(...) / ***PAV-PREFERRED-DEVICE**(...)

| *__PAV-PREFERRED-DEVICE__(...)

| | ,**PAV-PREFERRED-DEVICE** = ***UNCHA**NGED / ***NO** / ***Y**ES

. . .

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 for I/Os in VM2000 operation.

PAV-PREFERRED-DEVICE = *UNCHANGED
The current setting remains unchanged.

PAV-PREFERRED-DEVICE = *YES
The base or alias device is preferred for I/Os in VM2000 operation.

PAV-PREFERRED-DEVICE = *NO
The preference is canceled.

5.7.5 MODIFY-IO-OPTIONS (Modify settings for I/O devices)

Using the MODIFY-IO-OPTIONS command the system support can modify the following I/O settings of the input /output devices:

- Timeout value for I/O monitoring

  Changes to timeout values are suggested for specific actions that might cause time delay in the execution of an I/O request (e.g. during the online update of RAID system's firmware). Upon completion of these actions, the default timeout values should be restored.

- FastDPAV options for disks, if FastDPAV (s. "Parallel access volume (PAV, SU /390)") is supported in the system

  In connection with FastDPAV, an additional performance improvement can be reached for disks assigned shared to the VM.
  Selecting a FastDPAV alias device as preferred device for an I/O request can increase the performance additively, as the alias devices are assigned to the VM exclusively, and therefore there is no hypervisor overhead for these devices.
  The setting of this option is common for all disks that are assigned shared to the respective system.

  x86 FastDPAV devices are not available on SU x86, where the inputs/outputs are automatically parallelized by X2000. The VM2000-specific operands for prioritizing FastDPAV devices in this BS2000 command have consequently no effect on SU x86.

In VM2000 operation, the command MODIFY-IO-OPTIONS enables the modification of the options in all guest systems that support this command, if it is entered in the monitor system.

MODIFY-IO-OPTIONS

UNIT = alphanum-name_2..2 / x-text_4..4 / ***ALL** / ***CONTROLLER**(...) / ***DEVICE-RANGE**(...)

***CONTROLLER**(...)

| **NAME** = <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>

,**TIMEOUT** = ***UNCHANGED** / ***SYSTEM-DEFAULT** / <integer 16..86400>

,**FAST-DPAV** = ***UNCHANGED** / ***PARAMETER**(...)

***PARAMETER**(...)

| **PREFERRED-DEVICE** = ***BASE-DEVICE** / ***ALIAS-DEVICE**

,**SCOPE** = ***OWN-SYSTEM-ONLY** / ***VM2000-GLOBAL**

Description of the operands (VM2000-specific operands)

FAST-DPAV = ***UNCHANGED** / ***PARAMETER**(...)
Specifies settings for FastDPAV.

FAST-DPAV = ***PARAMETER**(...)

PREFERRED-DEVICE = *BASE-DEVICE / ***ALIAS-DEVICE**
Specifies the order of the selection of base and alias devices for the I/O execution of a VM-shared disk, if FastDPAV is supported in the system.
The setting is accepted independently of the devices specified in the `UNIT` operand.
If FastDPAV is not supported in the system, the command is rejected.

PREFERRED-DEVICE = *BASE-DEVICE
If no I/O is running on the base device, this one is chosen for an incoming I/O request. If an I/O is already running on the base device, a new I/O request is started on an alias device, provided there is one available. This is the system preset device selection.

PREFERRED-DEVICE = *ALIAS-DEVICE
If no I/O is running for the device (neither on the base nor on an alias), a new I/O request is started on an alias device, provided there is one available. If there is already an I/O request running for the device, a new one will be started on the base device, if there is no I/O running on it or if no (further) alias device is available.
This device selection mode can improve the I/O performance, if there are sufficient alias devices available. It may be unfavourable, if volumes with low I/O rate occupy the alias devices and volumes with higher I/O rate have no or too few alias devices at their disposal.

SCOPE = *OWN-SYSTEM-ONLY / ***VM2000-GLOBAL**
Specifies how the command is to be executed in VM2000 operation.

SCOPE = *OWN-SYSTEM-ONLY
The command is executed in the local system only.

SCOPE = *VM2000-GLOBAL
If the command is entered in the monitor system, the modification of the I/O options is enabled in all guest systems that support this command. If entered in another guest system, the command is rejected.

5.7.6 SHOW-SYSTEM-INFORMATION (Display system information)

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, BS2000 host name, VM identification, system name and origin of the system parameters at system startup)
- the VM2000 version
- information on the monitor system (BS2000 version and BS2000 host name)

The command supports structured output to S variables.

Format (VM2000-specific operands)

SHOW-SYSTEM**-INF**ORMATION
INFORMATION = ***ALL** / **. . .** / ***HSI-ATTR**IBUTES / ***SYS**TEM**-CONF**IGURATION / ***VM2000-VERS**ION / ***VM2000-MON**ITOR**-SYS**TEM / **. . .** ,**OUTPUT** = ***SYSOUT** / ***SYSLST** ,**SERVER-UNIT** = ***STD** / ***INITIAL** / ***CURRENT**

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              = <sysid>
                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 /...
                LIVE-MIG-COUNT     = <n>
```

Meaning of the output values

<vm-index>	In VM2000 operation, the VM index in the format <integer 1..99>; in native operation 0.
<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 (SU x86).

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. V12.0B

INFORMATION = *VM2000-MONITOR-SYSTEM
Provides information on the monitor system in VM2000 operation:

```
VM2000-MONITOR-  BS2000-VERSION [1]  = <bs2000-version>

SYSTEM:          HOST-NAME           = <host-name>
```

[1] The BS2000-VERSION replaces the former OSD-BC-VERSION in BS2000 from V21.0A.

Meaning of the output values

<bs2000-version>	Version in DOD format, e.g. V21.0B0000
<host-name>	BS2000 host name of the monitor system, e.g. D016ZE10

SERVER-UNIT = *STD
The currently valid settings of the BS2000 session should be used.

SERVER-UNIT = *INITIAL
Output are the data of the Server Unit on which the IPL was executed.

SERVER-UNIT = *CURRENT
Output are the data of the Server Unit on which the BS2000 session is currently running (if necessary after a Live Migration).

> The default value is usually `SERVER-UNIT=*INITIAL`, i.e. the SU on which the IPL was executed. Because of this, the SU does not change, even after a VM Migration. To output the (VM2000) data of the current SU, `SERVER-UNIT=*CURRENT` has to be specified explicitly!

Example (in the monitor VM)

```
/show-system-information information=*all ──────────────────────────────  (1)
%CONFIGURATION                      = 390SU-  710- 40
%CPU-ID-LIST :     ADR   0          = 3002900136000000
%                  ADR   1          = 3012900136000000
%                  ADR   2          = 3022900136000000
%HSI-ATT :         TYPE             = IX ──────────────────────────────────  (2)
%                  ASF              = YES
%                  OPERATION-MODE   = VM2000
%MEMORY-SIZE                        = 512 MB
%MINIMAL-MEMORY-SIZE                = 512 MB
%BS2000-ID :       NAME             = <name>
%                  VERSION          = V21.0B0000
%                  SERVICE-PACK     = SP 23.2
%                  CREATED          = <date><time>
%IOCONF-ID :       NAME             = SU710001
%                  VERSION          = V21.0A01
%                  CREATED          = <date><time>
%                  FORMAT           = IORSF01
%IPL-TIME                           = <date><time>
%SYSTEM-CONF :     SYSID            = 123 ──────────────────────────────────  (3)
%                  HOME-PUBSET      = SEMA
%                  HOST-NAME        = ABGSE211
%                  VM-INDEX         = 1
%                  VM-NAME          = MONITOR
%                  SYSTEM-NAME      = *NONE
%                  SYSPAR-BS2-SEL   = *BY-IOCONF-ID-NAME
%                  LIVE-MIG-COUNT   = 0 ──────────────────────────────────  (4)
%VM2000-VERSION                     = V12.0B ──────────────────────────────  (5)
%VM2000-MONITOR-   BS2000-VERSION   = V21.0B0000 ──────────────────────────  (6)
%   SYSTEM:        HOST-NAME        = ABGSE211
%SYSTEM-TIME-       ZONE             = +01:00
%   PARAMETER:     SEASON           = S
%                  SEASON-DIFFERENCE = 01:00
%                  PREV-CHANGE-DATE  = <date><time>
%                  NEXT-CHANGE-DATE  = <date><time>
%                  SYNCHRONIZATION   = SERVER-CONN-EXT-REF
% EPOCH                             = <epoch>
```

---

(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) Counter for the VM Migrations (Live Migration)

(5) Corresponds to the output for `INFORMATION=*VM2000-VERSION`

(6) 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 S variable	T	Contents	Cond.
. . .				
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 version of the monitor system (in VM2000 operation), otherwise *NONE	var(*LIST).VM-MONITOR.BS2000-VERSION	S	<c-string 10> *NONE	11

5.7.7 SHOW-DEVICE-CONFIGURATION (Display device 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 [13] 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**-CON**FIGURATION (VM2000 specific)

UNIT = ***STD** / ***SEL**ECT(...) / ***PUB**SET**-DEV**ICES(...) / **. . .**

***SEL**ECT(...)

| **CLASS** = ***ALL** / **. . .** / ***DEV**ICE(...)

| ***DEV**ICE(...)

| | **TYPE** = ***ALL** / <device>

| | ,**VM-ASSIGNMENT** = ***ANY** / ***NO** / ***Y**ES / ***BY-GUEST-POSSIBLE**

| ,**ATTR**IBUTE = **. . .**

. . .

***PUB**SET**-DEV**ICES(...)

| **PUB**SET = <cat-id 1..4> / ***BY-PUBRES-DEV**ICE(...)

| ***BY-PUBRES-DEV**ICE(...)

| | **UNIT** = <alphanum-name 2..2> / <x-text 4..4>

,**INF**ORMATION = ***STD** / **. . .** / ***VM2000**

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.

> **i** 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
If INFORMATION=*VM2000 is specified, 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 own VM but that can currently be assigned implicitly (using /ATTACH-DEVICE, for example).

...

INFORMATION = *VM2000
Information on (possible) assignment to own VM and on usage mode.

Output format:

```
    MNEM UN-CLASS UN-TYPE CONF-STATE   VM-ASSIGN      VM-USAGE
```

Example

```
/show-device-configuration unit=(d2,d3,d4,d5,d6,d7,p2,p3),
                           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
P2   DVC      HNC      ATTACHED     BY-GUEST       EXCL
P3   DVC      HNC      ATTACHED     BY-GUEST       EXCL

------------------------------------------------------------------------------------------------------------------------
```

(1) VM2000 information is requested for the devices specified

(2) See "Meaning of the output columns with INFORMATION=*VM2000" for the meaning of the output columns VM-ASSIGN and VM-USAGE

S variables

Output information	Name of S variable	T	Contents	Condition
Assignment mode or option of own VM	var(*LIST).VM-ASS	S	BY-GUEST BY-VM-ADM VM-INHIBIT DEV-INHIBIT ELSEWHERE BY-GUEST-POSSIBLE	INF=*VM2000
Usage mode in VM2000	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	Displays assignment mode or option for the own VM: BY-VM-ADM The device is assigned explicitly to the own VM. BY-GUEST The device is assigned implicitly to the own VM. BY-GUEST-POSS The device is not assigned to the own VM, implicit assignment is possible. In the following cases, the device is not assigned to the own VM and implicit assignment is not possible: VM-INHIBIT The VM is not authorized for implicit assignment at all. DEV-INHIBIT The VM is not authorized to assign this device implicitly. Exception for VM-INHIBIT / DEV-INHIBIT: Snap units (Snapsets) can nevertheless be assigned implicitly to a VM with the AUTO-SNAP-ASSIGNMENT privilege (see "Initializing a VM"). ELSEWHERE The device is assigned exclusively to another VM.

Keyword	Meaning
VM-USAGE	Displays the usage mode in which the device is assigned (depends on VM-ASSIGN). If VM-ASSIGN = BY-GUEST / BY-VM-ADM: EXCL The device is assigned exclusively to the own VM. SH(D) The device is assigned to the own VM for shared usage with direct I/O processing (only one VM sharer). SH(I) The device is assigned to the own VM for shared usage with indirect I/O processing. If VM-ASSIGN = BY-GUEST-POSS: 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). SH(D) The device is assigned to another VM for shared usage with direct I/O processing (only one VM sharer). SH(I) The device is assigned to another VM for shared usage with indirect I/O procesing. NONE The device is not assigned. If VM-ASSIGN = VM-INHIBIT / DEV-INHIBIT: EXCL The device is assigned exclusively to another VM. SH(D) The device is assigned to another VM for shared usage with direct I/O processing (only one VM sharer). SH(I) The device is assigned to another VM for shared usage with indirect I/O processing. NONE The device is not assigned. If VM-ASSIGN = ELSEWHERE: EXCL The device is assigned exclusively to another VM.

6 VM2000 management

This chapter describes the management functions of VM2000. The management functions are realized by 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".

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: paragraph „Logging” in sections "Administration dialogs from privileged user tasks" and "Operating the virtual machines (ADMIN dialog via $VMCONS)".

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 procedure

A general example for CPU management is provided in the section "Example of CPU management".

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 Unit
- Managing the extra CPUs of the Server Unit

> **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

The real CPUs of a Server Unit are subdivided into:

- CPUs which are available in VM2000 operation
  The number depends on the server model and the licenses, see also "Implementation on SU x86".

  These CPUs are subdivided into:

  - Real normal CPUs (type of CPU: `NORMAL`)
  - Real extra CPUs (type of CPU: `EXTRA`)
  - Real hot spare CPUs (type of CPU: `SPARE`, SU /390)

  VM2000 manages real normal and extra CPUs in the CPU pools, see section "Managing CPU pools". 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**.

  390 When the monitor system starts up, **all** available and operable real (normal) CPUs of SU /390 are added to the standard CPU pool and started.

  If hot spare CPUs or extra CPUs are available, these are placed in a standby state at startup. 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 carrier system (domain 0)
    (type of CPU: `FOREIGN`, SU x86)
  - Real CPUs which are unused in this session, e.g. because no license exists or real cold spare CPUs
    (type of CPU: `UNUSED`, SU x86)

  x86 If more than 64 CPUs are installed, real CPUs of the `UNUSED` type are no longer displayed in the VM2000 information commands.

Real CPUs are in one of the following states:

ATT The CPU is attached. This state is attained after

- successful startup
- successful `/ATTACH-VM-RESOURCES`.

DET The CPU is detached. This state 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).
390 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

390 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 state 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`

x86 On SU x86 these commands display **all** real CPUs of the server unit, i.e. also CPUs of type `FOREIGN` or `UNUSED`. The CPUs of the type `FOREIGN` are always in the ATT state, the CPUs of the type `UNUSED` are in the DET state.

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 (depending on the architecture of the Server Unit), in other words with a maximum of 32 virtual CPUs (CPU 00 through CPU 1F), see "Initializing a VM".
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 "Increasing capacity with extra CPUs".

> ¡ The multiprocessor level for a VM can also be entered in the VM definition with `/CREATE-` or `/MODIFY-VM-DEFINITION`. Detailed information can be found in the section "Working with VM definitions".

Once a VM has been initialized, 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 "Managing CPU pools").
>
> 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**.

The multiprocessor level of the monitor VM is set when VM2000 is installed (see "Configuration using the SE Manager (SU x86)").

> 390 If hot spare CPUs are available on SU /390, each VM is assigned not only virtual normal CPUs but also virtual spare CPUs, see "High availability with hot spare CPUs (SU /390)".

Virtual CPUs are assigned one of the following states:

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)`
- Migration of a VM with virtual CPUs in BLOCK state to the target SU, on which the corresponding CPU pool has more real CPUs attached than the CPU pool on the source SU (`/MIGRATE-VM`)

IDLE The CPU is in the interruptible wait state.

INIT The CPU is initialized (VM in INIT-ONLY state or (on SU x86) the state of a virtual CPU up to the automatic attachment when starting up in the BS2000 guest system).

WAIT The CPU has been stopped by VM2000 (VM in IN HOLD(WAIT) state).

HALT x86 The CPU has been halted by X2000 (short-term transitional state or error).

STOP The CPU has been stopped (stop by the hardware). This state is attained after

- successful `/DETACH-DEVICE`
  (VM not overdimensioned, see "Number of attached real and virtual CPUs in the CPU pool")
- the CPU pool of an active VM has been switched without automatic attachment of virtual CPUs (for CPUs in BLOCK state)
  `(/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 state)
  `(/SWITCH-VM-CPU ...,TARGET-CPU-POOL=*NONE)`
- 390 (machine check) error

In these cases the CPU can be attached again during ongoing operation.

BLOCK The CPU is “blocked” by VM2000. This state is reached after the following actions (in all cases the VM was overdimensioned beforehand, see "Number of attached real and virtual CPUs in the CPU pool"):

- Startup of the guest system
- successful `/DETACH-DEVICE`
- 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
- to the target SU on which the corresponding CPU pool has less real CPUs attached than the CPU pool on the source SU. Because of this, some virtual CPUs of the migrated VM have to be automatically detached (`/MIGRATE-VM`).

In these cases the CPU in the guest system cannot be attached again.

SLEEP 390 The hot spare CPU is ready but sleeping.
This VM currently contains only one attached normal virtual CPU.

This state can also occur temporarily during a CPU reconfiguration.

OFF 390 The CPU is not ready (offline). This state 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 startup of the guest system

The VM2000 administrator can use `/SHOW-VM-RESOURCES` and the `INFORMATION=*CPU,VM-IDENTIFICATION=...` operand to obtain information on the state of the specified VMs' virtual CPUs.

The VM administrator can use `/SHOW-VM-ATTRIBUTES` and the `INFORMATION=*CPU` operand to obtain information on the state 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 "Scheduling procedure"). The virtual CPU selected is then started up on the real CPU.

390 On SU /390, scheduling is implemented by the VM2000 hypervisor.

x86 On SU x86, 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 "Automating VM2000 and VM administration".

Failure of real CPUs

390 On SU /390, 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 CPUs, see "High availability with hot spare CPUs (SU /390)".

CPUs in WFM state (SU /390 with “hot spare CPU”, see "Real CPUs") must be made available again by customer support. It is then necessary to reboot the Server Unit.

If a virtual CPU was assigned to the real 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.

x86 On SU x86, 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 "Implementation on SU x86") are configured in such a way that the standard CPU pool as far as possible contains the number of BS2000 CPUs envisaged for this Server Unit.

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".

If a detached real CPU is to be made available again, the VM2000 administrator can attach it again by means of `/ATTACH-VM-RESOURCES`.

> i 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). 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” [18]
- 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`)
- VM Migration with automatic attachment or detachment of virtual CPUs

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 state of the virtual CPUs.

6.1.1.4 High availability with hot spare CPUs (SU /390)

Hot spare CPUs are redundant, fully operable CPUs 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.

x86 Hot spare CPUs are not available on SU x86.
Cold spare CPUs, which (after a CPU has failed) are placed in operation after a Server Unit has been rebooted, can be configured there. The former CPU capacity is then available after the restart.

Selected SU /390 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.

> If the total number of virtual normal CPUs and virtual spare CPUs is greater than 16 (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 (state OFF), since they have at least one additional normal CPU available for failure detection.

Monoprocessor guest systems connect a spare CPU so that the operating system can detect the failure of its normal CPU and respond to it (state SLEEP). The number and state of virtual spare CPUs is displayed in the VM specific part when `/SHOW-VM-ATTRIBUTES INFORMATION=*STD/*CPU` or `/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 grade of the Server Unit remains unchanged by this. The spare CPU is automatically added to the CPU pool to which the failed CPU belonged.

The defective CPU (state 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 becomes available again when the Server Unit is rebooted. After it is repaired, the service engineer makes the CPU available again. It must then be rebooted.

> ℹ Other failures of normal or extra CPUs (e.g. when a CPU gets hung) lead to the CPU being detached (state 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 Unit is restarted.

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 grade and power consumption of a VM are not changed by the failure.

Monoprocessor guest systems process the CPU error on the virtual spare CPU having been in state SLEEP. Monoprocessor 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 operable virtual CPU in the monitor system fails, the VM2000 hypervisor initiates a restart of the monitor system, provided the restart option is set,

6.1.1.5 Increasing capacity with extra CPUs

Extra CPUs are redundant, fully operable CPUs of a Server Unit 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 Server Units 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 Unit (see "Planning distribution of the CPU capacity to the VMs"), 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 remain logically disconnected when VM2000 operation is initialized and are added to the standard CPU pool. On SU /390, the number of available extra CPUs is indicated in message `VMS5053` when the monitor system is started up. The state 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.

390 In the event of the failure of extra CPUs on SU /390, available hot spare CPUs are attached automatically by VM2000.

x86 In the event of the failure of extra CPUs on SU x86, 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 Unit 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".

Simple example of a Server Unit with 2 CPUs (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

[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**

[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**

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" (Planning distribution of the CPU capacity to the VMs) 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 <= PROCESSOR <= 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 "Increasing capacity with extra CPUs", VM1 and VM2 therefore cannot be set up as monoprocessor 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 "CREATE-VM (Initialize a VM)".

The limit on the CPU utilization of a VM or VM group imposed by `MAX-CPU-UTILIZATION < 100` (see "Planning distribution of the CPU capacity to the VMs") continues to apply to the nominal CPU capacity of the Server Unit. 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 SU /390) 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.

- Static CPU pools
- Dynamic CPU pools
- Number of attached real and virtual CPUs in the CPU pool
- Creating a CPU pool
- Adding CPUs to a CPU pool
- Assigning a VM or VM group to a (different) CPU pool
- Canceling the assignment of a VM or VM group to a CPU pool
- Outputting information on a CPU pool
- Deleting a CPU pool
- Example of managing CPU pools

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 Units.

x86 Further static CPU pool is available on SU x86 (`*POOL0`), see section "Implementation on SU x86".
This is displayed in VM2000 information commands, but 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.

390 Spare CPUs are not assigned to a CPU pool on SU /390.

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 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 (SU /390)".

VMs and VM groups which are assigned to a CPU pool

After `VM2000 READY`, VMs and (on SU /390) 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".

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] CREATE-VM-DEFINITION [1, 3] MODIFY-VM-DEFINITION [1, 3]
Canceling the assignment of a VM or VM group to a CPU pool	DELETE-VM [4] DELETE-VM-DEFINITION [1, 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 SU /390

3 Command only for working with VM definitions

4 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 "Creating a CPU pool".

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)

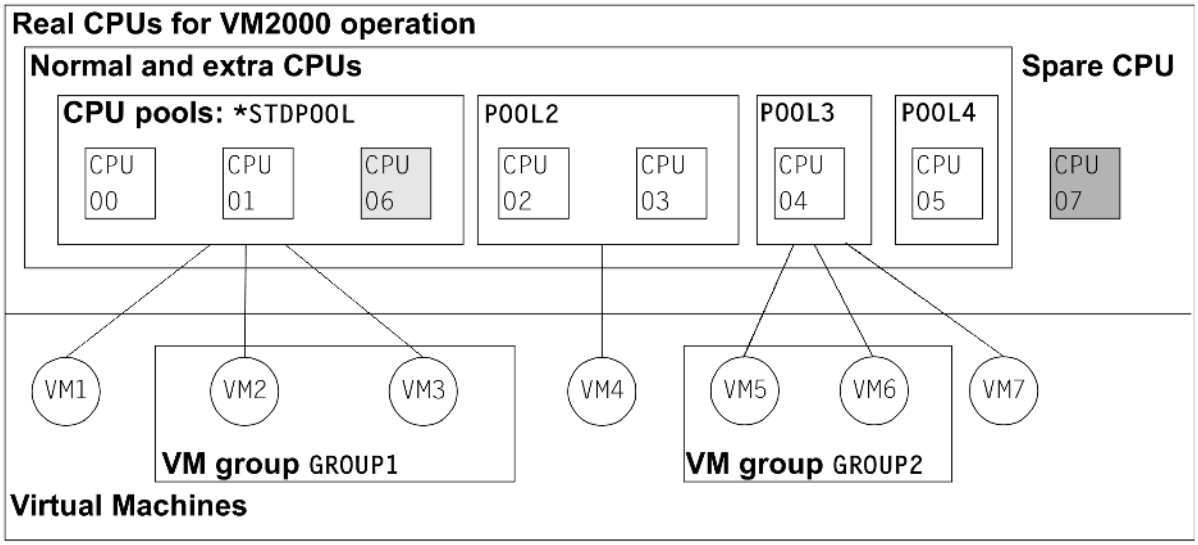


Figure 16: Schematic overview: Possible assignments of VMs to CPU pools

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 state RUNNING, SETUP(MIGR) or IN HOLD (WAIT).
- The CPU pool is assigned a VM in state INIT-ONLY or DOWN (SU x86).

The number of attached virtual CPUs of a VM in state RUNNING or IN HOLD (WAIT) 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 state RUNNING or IN HOLD (WAIT) is no longer fulfilled (overdimensioned VM), the VM will nevertheless continue to run with a too high multiprocessor level and the resultant disadvantages.

If the number of attached real CPUs is reduced, in most cases VM2000 guarantees that this principle will be fulfilled. 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 fulfilled.

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 removed, but a VM in state RUNNING, SETUP (MIGR) or IN HOLD (WAIT) is assigned to the CPU pool, or, on SU x86, also in state INIT-ONLY or DOWN (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 fulfilled 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 (state BLOCK). They can be attached in the guest system later only if further attached real CPUs are added to the CPU pool.

- A VM (state RUNNING or IN HOLD (WAIT)) or VM group (with VMs in state RUNNING or IN HOLD (WAIT)) is assigned to a different CPU pool (/ASSIGN-VM-(GROUP-)TO-CPU-POOL):

  If the principle is not fulfilled 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.
    (State of the virtual CPU in the information commands of VM2000: BLOCK).

- A VM in state RUNNING is being migrated and the CPU pool on the target SU has fewer real CPUs attached than the CPU pool on the source SU (/MIGRATE-VM):

  If the principle is not fulfilled in the new CPU pool, the command is executed if the CPU pool to which the migrated VM is to be assigned contains at least one attached CPU. In addition, VM2000 attempts to detach the excess CPUs (asynchronous, implicit /DETACH-DEVICE in the guest system for a corresponding number of virtual CPUs). Initially, these detached CPUs cannot be attached again by the guest system. They are identified accordingly by VM2000.
  (State 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 in the following cases:

  390 When the CPU pool is assigned a VM with an active guest system (message/return code VMS4617)

  x86 When the CPU pool is assigned a VM

  If the principle is no longer fulfilled for a VM in state RUNNING or IN HOLD (WAIT), 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 state BLOCK 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 a SU /390 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 paragraph "A real CPU is detached".

  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 itself is 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 fulfilled. Nevertheless, the VM2000 administrator should take the following into account:

- Attached real CPUs are added to a 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 state BLOCK for the relevant number of virtual CPUs of 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 state of the virtual CPUs of all VMs of the target 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 state BLOCK 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 (state RUNNING or IN HOLD (WAIT)) or a VM group (with VMs in state RUNNING or IN HOLD (WAIT)) 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 state BLOCK is canceled for a corresponding number of virtual 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).
- A VM in state RUNNING is being migrated und the CPU pool on the target SU has more real CPUs attached than the CPU pool on the source SU (/MIGRATE-VM):

  The command is executed. If the multiprocessor level of a VM is greater than the number of attached virtual CPUs, the state BLOCK is canceled for a corresponding number of virtual CPUs. In addition, VM2000 attempts to attach the virtual CPUs that were in state BLOCK (asynchronous, implicit /ATTACH-DEVICE in the guest system for these 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 Unit's real CPUs or the maximum possible number of virtual machines. The number of VMs depends on the architecture of the Server Unit, see "CREATE-VM (Initialize a VM)".

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.

x86 On SU x86, 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 the VM2000 operation belong to the standard CPU pool (*STDPOOL).

The VM2000 administrator can use /SWITCH-VM-CPU to remove 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 (SU /390) 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 (see "Initializing a VM") or VM group is initialized (using `/CREATE-VM-GROUP` (with the `CPU-POOL-NAME` operand)).

> ¡ `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used to enter the assignment of a VM to a CPU pool in its VM definition. Detailed information can be found in the section "Working with VM definitions".

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 "Planning distribution of the CPU capacity to the VMs".

6.1.2.7 Canceling the assignment of a VM or VM group to a CPU pool

The assignment of a VM to a CPU pool is also implicitly canceled by the VM administrator when terminating the VM (`/DELETE-VM`).

The assignment of a VM group to a CPU pool is also implicitly canceled by the VM2000 administrator when terminating the VM group (`/DELETE-VM-GROUP`).

> `/DELETE-VM-DEFINITION` cancels the assignment of the VM to a CPU pool entered in the VM definition.

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 condition is that no CPU and no VM or VM group are assigned to 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 SU /390. Asynchronous messages are marked with %.

```
/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                     NONE
SPARE                     04(DET)
/create-vm-cpu-pool cpu-pool-name=pcust1 ————————————————————————  (2)
VMS4601 CPU POOL 'PCUST1' CREATED
/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)
PCUST1     -   NORMAL   02(ATT),03(ATT)
/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'
/create-vm memory-size=512,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=fc1c)) ———————  (9)
VMS4005 DEVICE 'FC1C' ASSIGNED TO VIRTUAL MACHINE (3,C1SYS2)
VMS4005 DEVICE 'FC1D' ASSIGNED TO VIRTUAL MACHINE (3,C1SYS2)
VMS4220 PUBSET '4IVL' WITH PUBRES DEVICE 'FC1C' 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=fc1c,vm-id=c1sys2
%  VMS2023 CPU 00 OF VIRTUAL MACHINE (3,C1SYS2) STARTED
%  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
/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)
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 'FC1C' REMOVED FROM VIRTUAL MACHINE (3,C1SYS2)
VMS4009 DEVICE 'FC1D' REMOVED FROM VIRTUAL MACHINE (3,C1SYS2)
```

```
VMS4619 VIRTUAL MACHINE (3,C1SYS2) REMOVED FROM CPU POOL 'PCUST1'
VMS4024 VIRTUAL MACHINE (3,C1SYS2) TERMINATED
/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
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
```

---

(1) The command shows the initial situation for this example:
The data center has four real normal CPUs on this Server Unit. The existing spare CPU plays no role. A customer VM `C1SYS1` with the attributes shown is running. Time slicing (`TS`) is used for scheduling, see "Scheduling procedure". 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".

(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`.

(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 their virtual CPUs 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`. After detachment virtual CPUs 02 and 03 are still “blocked” (CPU status BLOCK).

(5) The information output for the CPU pools shows the distribution of real normal 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.

(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 "Scheduling procedure".

(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).

(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 "Scheduling procedure".

(14) The VM is terminated and removed from the customer CPU pool PCUST1.

(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 their virtual CPUs 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

In 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:

- **The CPU quota**

  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 when the VM is initialized (see "Initializing a VM") or specified in `/MODIFY-VM-ATTRIBUTES`.

  For VM groups (SU /390), 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 when the VM is initialized (see "Initializing a VM") or specified in `/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 **CPU capacity of the Server Unit**, which is referred to as the **effective CPU share of the VM (EFF-Q)**, see paragraph "Calculating the effective CPU share of a VM".

  > For the definition of the CPU capacity of a Server Unit, see paragraph "CPU capacity and CPU share of a VM".

  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 of the VM is specified in the `MAX-CPU-UTILIZATION` operand when the VM is initialized (see "Initializing a VM") or specified in `/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 paragraph "Calculating the effective CPU share of a VM".

  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 paragraph "Output of the current planned and measured values for the CPU share of the VMs".

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 utilization of the Server Unit. In the case of peak operation in all VMs, the effective CPU share of a VM is determined using the formulae in the paragraph "Calculating the effective CPU share of a VM". During offpeak 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 SU)	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 SU /390), 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 Unit 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 SU)	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 <= 100`). In this case, the maximum CPU utilization restricts the VMs which unintentionally take up a lot of CPU capacity. Nevertheless, the Server Unit 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 SU)	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, SU /390)

VM (mono SU)	CPU-QUOTA	MAX-CPU-UTILIZATION	Effective CPU share of the VM [1]
1	10	100	30 %
2	15	100	45 %
VM group `GROUP1`	25	25	(25 %) [2]
	MEMBER-CPU-QUOTA		
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 Unit 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 Unit's capacity!

> 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 Server Units 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 Unit or CPU pool to which the VM is assigned.

The **CPU capacity of a Server Unit** is the CPU capacity of the real (normal) CPUs which are available in VM2000 operation (see "Real CPUs"), irrespective of whether they are attached or detached. It is expressed in terms of the number of CPUs.

> Attached extra CPUs increase the Server Unit's CPU capacity temporarily (see "Increasing capacity with extra CPUs"). 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 Unit 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 Server Unit with 4 CPUs has a CPU capacity which is 50% of that of the Server Unit.

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 "Parameters for distributing the CPU capacity") are used to calculate the CPU share of a VM in the Server Unit's CPU capacity, see "Calculating the effective CPU share of a VM".

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 "Constraints for planning" from the values for the CPU quota and the maximum CPU utilization set by the VM2000 administrator.

Case 1: Not taking VM groups into account

The **effective CPU share of the VM (EFF-Q)** is calculated using the formula:

(F1) $$\text{EFF-Q} = \frac{\text{CPU-QUOTA of the VM} \times \text{CPU capacity of the CPU pool}}{\text{Sum of the CPU-QUOTAs of all VMs of the CPU pool}} \%$$

Example for a CPU pool with 2 normal CPUs and 4 available normal CPUs

	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) $$\text{EFF-Q} \leq \text{Maximum CPU utilization of the VM (by means of command)} \%$$

(N2) $$\text{EFF-Q} \leq \frac{\text{Number of real normal CPUs of the CPU pool} \times 100}{\text{Number of available real normal CPUs}} \%$$

(N3) $$\text{EFF-Q} \leq \frac{\text{Number of virtual CPUs of the VM (multiprocessor level)} \times 100}{\text{Number of available real normal CPUs}} \%$$

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) $$\text{EFF-Q} \leq \text{EFF-U} = \text{Minimum((N1),(N2),(N3))}$$

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) $$\text{EFF-Q} = \frac{\text{CPU-QUOTA of the VM} \times (\text{CPU capacity of the CPU-Pools} - \text{SUM(LIM)})}{\text{Sum of the CPU-QUOTAs of all unrestricted VMs of the CPU pool}} \%$$

where **SUM(LIM)** = 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 Unit'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 VM groups into account (SU /390)

The **effective CPU share of the VM** (**EFF-Q**) is calculated using the formula:

(F1a) $$\text{EFF-Q (VM or VM group)} = \frac{\text{CPU-QUOTA (VM or VM group)} \times \text{CPU capacity of the CPU pool}}{\text{Sum of CPU-QUOTAs of all single VMs and VM groups of the CPU pool}} \%$$

(F1b) **EFF-Q (VM in the VM group)** = $\frac{\text{MEMBER-CPU-QUOTA of the VM} \times \text{EFF-Q of the VM group}}{\text{Sum of the MEMBER-CPU-QUOTAs of all VMs in the VM group}}$ %

Example for a CPU pool with 2 normal CPUs on a Server Unit with 4 CPUs

		VM group GROUP1	
VMs	**VM1 (BI)**	**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 formulas (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 "CPU assignment in time slicing (SU /390)"

The effective CPU share of a VM may be restricted by the following constraints:

(N1a) **EFF-Q <= Maximum CPU utilization of the VM (by means of command) %**

(N1b) **EFF-Q <= Maximum CPU utilization of the VM group (by means of command) %**

(N2) **EFF-Q** <= $\frac{\text{Number of real normal CPUs of the CPU pool} \times 100}{\text{Number of available real normal CPUs of the server}}$ %

(N3) $$\text{EFF-Q} <= \frac{\text{Number of virtual CPUs of the VM (multiprocessor level) x 100}}{\text{Number of available real normal CPUs of the server}} \%$$

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) $$\text{EFF-Q} <= \text{EFF-U} = \text{Minimum((N1a),(N1b),(N2),(N3))}$$

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) $$\text{EFF-Q} = \frac{\text{CPU-QUOTA of the VM x (CPU capacity of the CPU pool - SUM(LIM))}}{\text{Sum of the CPU-QUOTAs of all unrestricted VMs of the CPU pool}} \%$$

where **SUM(LIM)** = 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 Server Unit with restricted VMs

		VM group GROUP1	
VMs	**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 formulas (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 Unit'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 "Parameters for distributing the CPU capacity". The relevant values are displayed using `/SHOW-VM-RESOURCES INFORMATION=*CPU`, see "SHOW-VM-RESOURCES (Output status of VM2000 resources)".

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 VM is migrated
- 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 Unit 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 "SHOW-VM-STATUS (Output VM2000 monitored data)"), 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 VMs in state RUNNING
- The active virtual CPUs

As with the idealized conditions, it is assumed that all VMs in the RUNNING status 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 "SHOW-VM-STATUS (Output VM2000 monitored data)".
The calculation is based on the formulae `EFF-Q` and `EFF-U`, which are shown on "Calculating the effective CPU share of a VM", 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 procedure

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 "Planning distribution of the CPU capacity to the VMs").

390 On SU /390, 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

x86 On SU x86, 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, SU /390)

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 (state RUNNING) 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-CPUS`, see "Initializing a VM").
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 "Initializing a VM". 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 "SHOW-VM-STATUS (Output VM2000 monitored data)".

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.

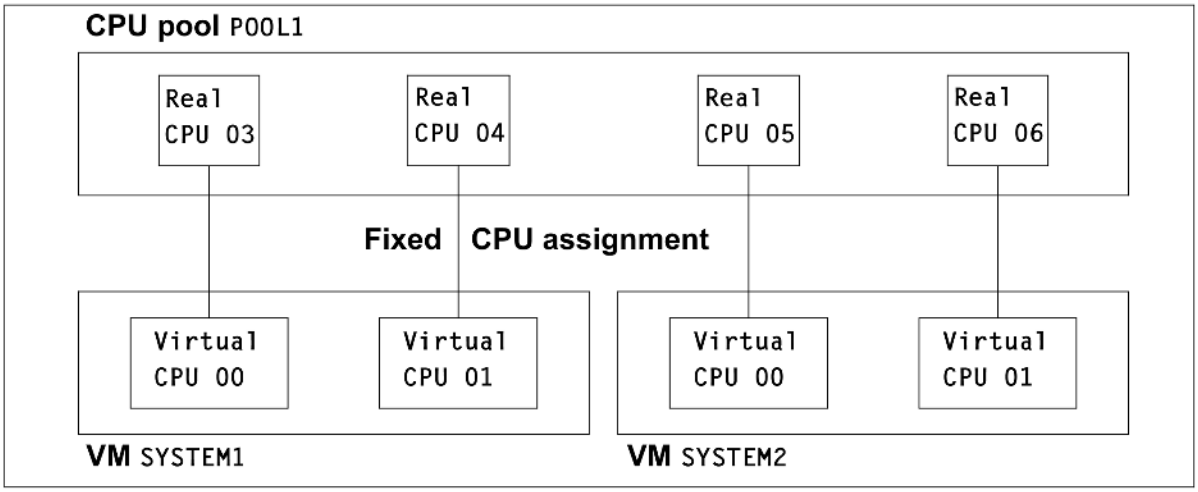


Figure 17: Example of fixed CPU assignment

In the example above, the CPU capacity of `POOL1` on a Server Unit with 8 CPUs 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 Unit), provided the maximum CPU utilization of the VM does not restrict this further.

6.1.4.2 CPU assignment in time slicing (SU /390)

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 (state RUNNING) 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) that is to run on a free real CPU of 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"):

- 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 recent past (CPU intensity)
  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 or a 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 or a 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 Unit 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 "Planning distribution of the CPU capacity to the VMs").

Example

Time slice size for a bi-VM with `EFF-Q=0.5` on a Server Unit with 4 CPUs:

```
0,5 * 4 (real CPUs) / 2 (virtual CPUs) = 1,0 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 VM-specific information block, output field `TIME SLICE DEF`, see "SHOW-VM-STATUS (Output VM2000 monitored data)".

It is the task of the VM2000 administrator to ensure that the effective (EFF-Q) and consequently the current (CUR-Q) CPU share of the VM is large enough to permit the VM to obtain control of the CPU sufficiently frequently, generally once a second (see "Planning distribution of the CPU capacity to the VMs").

6.1.5 Example of CPU management

This example explains the various outputs for CPU management in the various VM2000 information commands on SU /390.
Asynchronous messages are marked with %.

An example relating specifically to the management of the CPU pool is provided in section "Example of managing CPU pools".

```
/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)
              ———————————————————————————————————————————————————————— (3)
VM-ID         SCHED ACT-I   VIRT CPUS
 1 MONITOR     DC     NO     00(RUN),01(SLEEP) ———————————————————————   (4)
 2 SYSTEM2     -      NO     00(INIT),01(INIT),02(OFF)
 3 SYSTEM3     -      NO     00(INIT),01(OFF)
 4 SYSTEM4     DC    AT-DC   00(RUN),01(RUN),02(OFF) ————————————————   (5)
CPU-ATTR      CPU-POOL      REAL CPUS
NORMAL        *STDPOOL      00(ATT),01(ATT) ——————————————————————   (6)
              EXCLPOOL      02(ATT),03(ATT)
EXTRA                       NONE
SPARE                       04(DET)
/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)
/start-vm ipl-unit=fc1a,vm-id=system2,information-byte=*dialog ——————   (9)
% VMS2023 CPU 00 OF VIRTUAL MACHINE (2,SYSTEM2) STARTED
/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(INIT),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                     NONE
SPARE                     04(DET)
/show-vm-status information=*all,period=*recent(seconds=60) ———————— (11)
/SHOW-VM-STATUS DATA OUTPUT: (PERIOD=    61.844 SEC)
CPU TIMER INFORMATION FOR REAL CPU(S) ———————————————————————————— (12)
 CPU-POOL  REAL      HPV-ACTIVE          HPV-IDLE   ALL-VMS-ACTIVE
           CPU       TIME/    %       TIME/    %       TIME/    %
 *STDPOOL           0.017/ 0.01    123.625/99.95      0.046/ 0.04
            00      0.016/ 0.03     61.784/99.90      0.044/ 0.07
            01      0.001/ 0.00     61.841/  100      0.002/ 0.00
 EXCLPOOL           0.014/ 0.01      0.000/ 0.00    123.674/99.99
            02      0.007/ 0.01      0.000/ 0.00     61.837/99.99
            03      0.007/ 0.01      0.000/ 0.00     61.837/99.99
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       0.043/  0.01
   1 MONITOR        20.00   25.00   25.00       0.041/  0.02
              00                                0.041/  0.01
  EXTGROUP(G)       80.00   25.00   25.00       0.002/  0.00
   2 SYSTEM2        70.00   25.00   25.00       0.002/  0.00
              00                                0.002/  0.00
 EXCLPOOL(P)               50.00   50.00     123.653/ 49.98
   4 SYSTEM4         1.00   50.00   50.00     123.653/ 49.99
              00                               61.777/ 24.97
              01                               61.876/ 25.01
CPU COUNTER INFORMATION FOR REAL CPU(S) ——————————————————————————— (14)
 CPU-POOL    REAL    I-O INT.  INTERCEPT.   EMULATION     SCHEDULE
             CPU    COUNT/  %   COUNT/  %   COUNT/  %    COUNT/  %
 *STDPOOL     00      281/100    1153/ 47       4/100     1153/ 47
              01        0/  0      61/  2       0/  0       61/  2
 EXCLPOOL     02        0/  0     625/ 25       0/  0      625/ 25
              03        0/  0     624/ 25       0/  0      624/ 25
CPU COUNTER INFORMATION FOR VM(S)
 CPU-POOL    VIRT    I-O INT.  INTERCEPT.   EMULATION     SCHEDULE
 VM-ID       CPU    COUNT/  %   COUNT/  %   COUNT/  %    COUNT/  %
 *STDPOOL
  1 MONITOR   00      281/100    1153/ 47       4/100     1153/ 47
  2 SYSTEM2   00        0/  0      61/  2       0/  0       61/  2
 EXCLPOOL
  4 SYSTEM4   00        0/  0     624/ 25       0/  0      624/ 25
              01        0/  0     625/ 25       0/  0      625/ 25
SCHEDULE INFORMATION FOR REAL CPU(S) ——————————————————————————— (15)
 CPU-POOL    REAL      <-- TIME SLICE  -->  NON-AFFIN SCHED
             CPU        AVG %RUNOUT %SHORT    %IDLE %LOAD
 *STDPOOL     00        0.03      0     92  DEDICATED SCHED
              01        0.03      0    100  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  0.03      0     92  DEDICATED SCHED    --
  2 SYSTEM2   00  8.00  0.02      0    100  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
%  VMS2050 GUEST SYSTEM ON VM (2,SYSTEM2) READY
/show-vm-status information=*timer,period=*recent(seconds=180) ———————— (18)
/SHOW-VM-STATUS DATA OUTPUT: (PERIOD=   184.745 SEC)
CPU TIMER INFORMATION FOR REAL CPU(S)
 CPU-POOL  REAL      HPV-ACTIVE          HPV-IDLE   ALL-VMS-ACTIVE
           CPU       TIME/    %       TIME/    %       TIME/    %
 *STDPOOL          2.004/ 0.54    356.243/96.41     11.243/ 3.04
            00     1.113/ 0.60    177.743/96.21      5.889/ 3.19
            01     0.891/ 0.48    178.500/96.62      5.354/ 2.90
 EXCLPOOL          0.044/ 0.01      0.000/ 0.00    369.446/99.99
            02     0.022/ 0.01      0.000/ 0.00    184.723/99.99
            03     0.022/ 0.01      0.000/ 0.00    184.723/99.99
CPU TIMER INFORMATION FOR VM(S)
 CPU-POOL
  VM-GROUP   VIRT   CPU-Q   CUR-Q   CUR-U        VM - ACTIVE
  VM-ID      CPU                                 TIME/     %
 *STDPOOL(P)               50.00   50.00      11.241/  1.52
#  1 MONITOR        20.00   10.00   25.00       0.161/  0.02 ———————————— (19)
            00                                  0.161/  0.02
# EXTGROUP(G)       80.00   40.00   40.00      11.080/  1.50
#  2 SYSTEM2        70.00   40.00   40.00      11.080/  1.50 ———————————— (20)
            00                                  9.889/  1.33
*           01                                  1.191/  0.16
*           02                                  0.000/  0.00
 EXCLPOOL(P)               50.00   50.00     369.377/ 49.98
   4 SYSTEM4         1.00   50.00   50.00     369.377/ 49.98
            00                                184.639/ 24.98
            01                                184.738/ 24.99
/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(INIT),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                      NONE
SPARE                      04(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
 1 MONITOR   *STDPOOL           20.00/ 25.00 100.00/ 25.00  1(+1) ──── (23)
 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)
... (Ausgabe gekürzt)
/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=   307.707 SEC)
CPU TIMER INFORMATION FOR REAL CPU(S)
 CPU-POOL  REAL      HPV-ACTIVE         HPV-IDLE   ALL-VMS-ACTIVE
           CPU       TIME/    %       TIME/    %       TIME/    %
**STDPOOL           2.342/ 0.38    598.897/97.32     14.175/ 2.30
            00      1.277/ 0.42    299.216/97.24      7.214/ 2.34
            01      1.065/ 0.35    299.681/97.39      6.961/ 2.26
*EXCLPOOL           0.179/ 0.03    140.206/22.78    475.029/77.19
            02      0.088/ 0.03     70.114/22.79    237.505/77.19
            03      0.091/ 0.03     70.092/22.78    237.524/77.19
CPU TIMER INFORMATION FOR VM(S)
 CPU-POOL
  VM-GROUP   VIRT   CPU-Q   CUR-Q   CUR-U        VM - ACTIVE
  VM-ID      CPU                                 TIME/     %
**STDPOOL(P)               50.00   50.00       0.292/  0.02
#  1 MONITOR       20.00   25.00   25.00       0.292/  0.02
             00                                0.292/  0.02
*EXCLPOOL(P)               50.00   50.00     488.993/ 39.72
```

```
#   4 SYSTEM4          1.00   25.00   50.00     473.893/ 38.50
                00                                236.942/ 19.25
                01                                236.951/ 19.25
# EXTGROUP(G)        1.00   25.00   40.00      15.100/  1.23
#   2 SYSTEM2       70.00   25.00   40.00      15.100/  1.23
                00                                 11.799/  0.95
*               01                                  3.301/  0.26
*               02                                  0.000/  0.00
```

---

(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 section "Planning distribution of the CPU capacity to the VMs":

- CPU capacity of `*STDPOOL` and `EXCLPOOL`: 50% each (two of four real CPUs each, see (6))
- For the VMs and VM groups in the standard CPU pool:
  - Monitor VM: `EFF-Q = 20*50/100 = 10%` according to the formula (F1a)
    `EFF-U = 1*100/4 = 25%` according to the constraint (N3)
  - VM group `EXTGROUP`: `EFF-Q = 80*50/100 = 40%` acc. to the formula (F1a)
    `EFF-U = 40%` by means of command
    - `SYSTEM2`: `EFF-Q = 70*40/100 = 28%` according to the formula (F1b)
      `EFF-U = 40%` according to the constraint (N1b)
    - `SYSTEM3`: `EFF-Q = 30*40/100 = 12%` according to the formula (F1b)
      `EFF-U = 20%` according to the constraint (N1a)
- For the VM in the CPU pool `EXCLPOOL`:
  - `SYSTEM4`: `EFF-Q = 1*50/1 = 50%` according to the formula (F1)
    `EFF-U = 2*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 is 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 (state ATT) each are contained in the standard CPU pool and in the CPU pool EXCLPOOL. The CPU capacity of *STDPOOL and EXCLPOOL is 50% each (2 of 4 real normal CPUs of the Server Unit, according to the formula in section "Planning distribution of the CPU capacity to the VMs"). The detached extra and spare CPUs (DET status) are not relevant for this example.

(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 Unit'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.

(10) SYSTEM2 has been set up as a biprocessor system and initially starts up on the virtual CPU 00 (IPL-CPU, state RUN). The virtual CPU 01 is still in SVP stop (state STOP). Furthermore, VM2000 has also assigned the virtual spare CPU 02 to the VM. It is not operable (state OFF) 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

(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 "Planning distribution of the CPU capacity to the VMs"). These differ from the planned values under idealized conditions (cf. (3)) as only 4 of 6 normal CPUs and only 3 of 4 VMs are active. Each virtual CPU thus runs on a real CPU of the Server Unit.
- 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

(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.

(17) The virtual spare CPU 02 is briefly started and then stopped again.

(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, is assigned all the VM group's share of the Server Unit's CPU capacity while SYSTEM3 is not active. * indicates rows which can contain falsified monitored data for ACTIVE times (the virtual CPUs 01 and 02 of the VM SYSTEM2 have been attached and detached in the monitoring interval).

(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.

(24) The changes become visible with these commands, too. See also the explanations for (19) and (20).

Note

The CPUs in CPU pool EXCLPOOL are no longer running in the scheduling procedure with a dedicated CPU assignment. Consequently, attribute VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS for VM SYSTEM4 has no effect any longer.

6.2 Reconfiguring Channels, controllers and paths (SU /390)

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.

x86 No channels are available on SU x86. 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. If 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.

x86 Global reconfiguration of hardware units is not available on SU x86. 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.

ℹ 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 "ATTACH-DEVICE (Attach hardware units)"):

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 section "VMGIORAL system parameter").

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** (state offline) after its successful detachment in the monitor system (/DETACH-DEVICE) is executed by VM2000 depending on the state 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 via global reconfiguration). In this case, the state of the channel changes to offline. Otherwise, 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 state of the channel changes to offline.

A channel whose state is offline can no longer be addressed or attached by a guest system.
Moreover, a channel whose state is offline is not connected at the hardware level (its state 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.

> **i** 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 "Initializing a VM").

If SCOPE=*VM2000-GLOBAL is specified, it is rejected in a guest system with message NKR0178.

6.3 Main memory management and reconfiguration

Main memory management in VM2000 includes

- Management of the main memory
- Reconfiguration of the main memory

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 [16].

6.3.1 Managing main memory

390 On SU /390, VM2000 manages the Server Unit'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.

x86 When a VM is initialized on SU x86, a share of the Server Unit's main memory is made available to the VM, see "Implementation on SU x86".

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 "Accounting record for a VM").

6.3.1.1 Managing main memory on SU /390

Main memory on SU /390 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 "Setting the VM2000 parameters in the parameter service"). 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 512 Mbytes. The main memory size of the monitor VM is defined in the configuration file for the monitor VM (see `/CREATE-VM` in section "Configuration via the VM2000 configuration file (SU /390)").

This is followed by the main memory of the other VMs. The size of the main memory of the VMs is defined when the VM is initialized (see "Initializing a VM").

> ¡ `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used to enter the size of the main memory for a VM in its VM definition. Detailed information can be found in the section "Working with VM definitions".
>
> It is recommended that the remaining main memory (without the VM2000 hypervisor and monitor VM) is distributed according to the expected CPU shares of the VMs, analogously to the calculations on "Planning distribution of the CPU capacity to the VMs". 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.

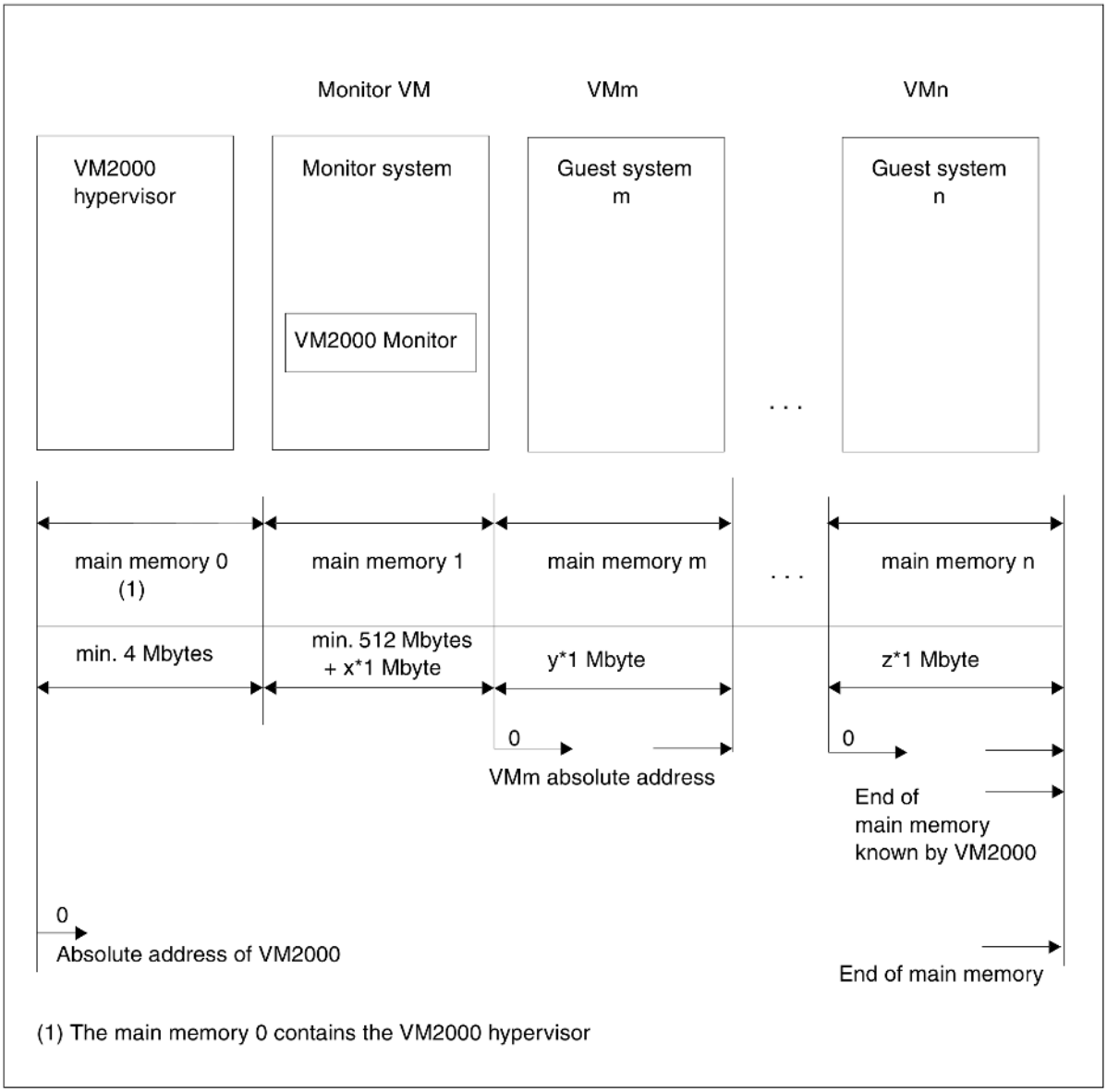


Figure 18: Distribution of main memory (SU /390)

Assigning main memory to a VM

The size of the main memory for the VM and its position in main memory is defined using the `MEMORY-SIZE, MIN-MEMORY-SIZE` and `LOCATION` operands when the VM is initialized (see "Initializing a VM"). If the location is not specified, VM2000 selects a suitable area.

> `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used to enter the size of the main memory of a VM in its VM definition. The assignment of main memory in `/ACTIVATE-VM-DEFINITION` then runs as with `/CREATE-VM`. Detailed information can be found in the section "Working with VM definitions".

When assigning free main memory space, VM2000 selects an area with the smallest “wastage” according to the specifications when the VM is initialized (see "Initializing a VM"). If no suitable area can be found, VM2000 rejects the command.

By reinitializing the VM, the VM administrator must then attempt to bring the size and/or 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	VM-absolute address decimal (Mbytes)
0 16	VM2000 hypervisor	
16 528	Monitor VM	0 512
528 1024	(Free main memory)	
1024 2048	VM2	0 1024
2048 4096	VM3	0 2048

Figure 19: Example: Distribution of main memory of 4 Gbytes among three VMs (SU /390)

6.3.1.2 Managing main memory on SU x86

On SU x86, 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 SU /390 - be characterized with the start address and the size of the main memory.

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 using the `MEMORY-SIZE, MIN-MEMORY-SIZE` and `MAX-MEMORY-SIZE` operands when the VM is initialized (see "Initializing a VM").

> `/CREATE-` or `/MODIFY-VM-DEFINITION` can be used to enter the size of the main memory of a VM in its VM definition. The assignment of main memory in `/ACTIVATE-VM-DEFINITION` then runs as with `/CREATE-VM`. Detailed information can be found in the section "Working with VM definitions".

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 "SHOW-VM-ATTRIBUTES (Output attributes of a VM)".
The main memory of a VM must be at least 1024 Mbytes large, see "Initializing a VM".

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 "SHOW-VM-RESOURCES (Output status of VM2000 resources)".

The size of the main memory of the monitor VM is defined when VM2000 is installed, see section "Configuration using the SE Manager (SU x86)". Its minimum size is 1024 Mbytes. The standard size of the monitor VM's main memory is 1024 Mbytes. This is the recommended value when the monitor VM is used only for VM2000 administration.

Releasing main memory

On SU x86 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

In 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 (SU x86) and in VM2000. It can be carried out for the monitor system as well.

The `MEMORY-SIZE` operand is used to dimension the main memory of the VM when the VM is initialized. The `MIN-MEMORY-SIZE` and `MAX-MEMORY-SIZE` (SU x86) operands 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 "Initializing a VM").

Main memory reconfiguration is possible at all times for inactive guest systems (with the status INIT-ONLY or DOWN). 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) or ...(MIGR) states.

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 (SU x86) and in VM2000, the process can be time consuming. The `/MOVE-VM` (SU /390) 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 "Monitoring the reconfiguration of a main memory with job variables"). 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 "Automating VM2000 and VM administration".

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 "Automating VM2000 and VM administration".

In the case of main memory expansion, an accounting record is written for the VM in all cases.

x86 On SU x86, aborting a main memory reconfiguration with /START-VM, /RESET-VM or /DELETE-VM should be avoided.
Reason: The reconfiguration will only be aborted in BS2000 but continued in the firmware and the carrier system. A BS2000 IPL with /START-VM may be delayed until the reconfiguration in the carrier system is complete.

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.

x86 On SU x86, 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. 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 (SU x86), 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.

390 On SU /390, the new main memory size may be below the MIN-MEMORY-SIZE of the VM concerned only if the VM is in state INIT-ONLY or DOWN. In this case, the value for the VM’s MIN-MEMORY-SIZE is assigned the value of NEW-MEMORY-SIZE.

x86 On SU x86, 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 (SU x86).

i 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 in VM2000, see "Monitoring a VM with job variables (MONJV)".

MONJVs can also be used to monitor a VM and during a VM Migration, see "Monitoring a VM with job variables (MONJV)" and "Monitoring a VM Migration with job variables (MONJV)".

For an example of how to use a MONJV to monitor the reconfiguration of a main memory, see below.

Beginning and end of monitoring

The MONJV used to monitor a main memory reconfiguration is created at the start of the main memory reconfiguration with `/EXTEND-VM-MEMORY` or `/REDUCE-VM-MEMORY` by specifying the `MONJV` operand. 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 paragraph with this name in section "Monitoring a VM with job variables (MONJV)".

Location of the MONJV

See the paragraph with this name in section "Monitoring a VM with job variables (MONJV)".

Values for the MONJV

A MONJV for a main memory reconfiguration 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 during the main memory reconfiguration.

Byte	Meaning / possible values
1-3	Monitoring status (left-aligned, padded with a space): • `$S` : Reconfiguration of main memory started (Started) • `$R` : Reconfiguration of main memory running (Running) • `$T` : Reconfiguration of main memory successfully terminated (Terminated) • `$AE` : Reconfiguration of main memory aborted with errors (e.g. due to cancellation of running reconfiguration of main memory, see above) • `$AF` : Reconfiguration of main memory could not be initiated
4	Reserved, has the value zero (0)
5-8	Reserved, contains spaces
9-12	Catalog ID of the home pubset of the monitor system
13-16	Reserved, contains spaces
17	Type of MONJV: character `V` (VM2000)
18-20	Current system sequence number (`<integer 1..999>`)
21-36	Beginning of monitoring (opening the MONJV) in UTC time (format: `yyyy-mm-ddhhmmss`)
37-70	Reserved
71-78	Name of VM (`<name 1..8>`, not for status `$S` and `$AF`)
79-81	Index of VM printable (`<name 3..3>`, e.g. `002`)
82-86	Memory size of VM in Mbytes (`<integer 1..32767>`, for `$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

[1] Specification of the memory size modulo 100 000. The complete values are output from byte 97 onward.

Displaying the MONJV

See the paragraph with this name in section "Monitoring a VM with job variables (MONJV)".

Behavior when the system is terminated abnormally in the monitor system

See the paragraph with this name in section "Monitoring a VM with job variables (MONJV)".

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 "Automating VM2000 and VM administration".

```
/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 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,MONJV=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
/JVRES = JV('JV.VMMEM.VM02')
/IF (SUBSTRING(JVRES,1,2) = '$A')
/ TXT = 'REDUCE-VM VM-ID=2 TERMINATED ABNORMALLY'
/ GOTO LABEL=ERROR
/END-IF
/"** EXTEND MEMORY OF VM3 ********"
/EXTEND-VM-MEMORY VM-ID=3,NEW-MEM-SIZE=*MAXIMUM,MONJV=JV.VMMEM.VM03
/IF-CMD-ERROR
/ TXT = 'ERROR IN EXTEND-VM VM-ID=3'
/ GOTO LABEL=ERROR
/END-IF
/"** SYSTEM MUST WAIT FOR END OF MEMORY CONFIGURATION  "
/WAIT-EVENT *JV(((JV.VMMEM.VM03,1,2) = '$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
/"** NORMAL 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
```

6.4 Peripheral device management

The same device peripherals can be operated in native mode and in 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".

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 in BS2000 is provided in the “Device management” chapter of the “Introduction to System Administration” [2].

Information on managing individual devices under VM2000 is provided in the section "Assigning and removing devices of a VM".

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. 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 in BS2000 is provided in the “Pubset-Management” chapter of the “Introduction to System Administration” [2].
The product HIPLEX MSCF is required for shared pubsets and XCS pubsets, see the “HIPLEX MSCF” manual [10].

In the device management and information commands of VM2000, pubsets can be addressed by means of their catalog ID or their pubres 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". 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".

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 "Explicit device assignment and removal")
- 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 "Assignment sets, implicit device assignment and release")

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.

390 For SU /390 this is implemented in the configuration file for the monitor VM

x86 For SU x86 this is implemented via the SE 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 "Explicit device assignment and removal"
- by a guest system (implicit release of implicitly assigned devices using BS2000 command `/DETACH-DEVICE`, see "Assignment sets, implicit device assignment and release"

Device name

Each device is referred to in the VM2000 commands by its **mnemonic device name**, analogously to the BS2000 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 "Explicit device assignment and removal").

Accounting for devices

The use of devices is displayed in the VM2000 accounting record for devices (see "Accounting record for devices").

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

> ¡ Devices for a VM can be entered in the VM definition with /CREATE- or /MODIFY-VM-DEFINITION. Detailed information can be found in the section "Working with VM definitions".

When the VM is assigned a device by the **VM2000 administrator**, the **VM administrator** receives an information message about every assigned device.

390 Alias devices (PAV/DPAV) and CRD devices cannot be assigned explicitly.

After devices are assigned explicitly 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** devices of a VM on which the guest system is still active is 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.

390 Alias devices (PAV/DPAV) and CRD devices cannot be explicitly removed.

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 devices to the VM themselves (implicitly), provided the requirements 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 (see "Dynamic I/O configuration modification").

390 The default value for the ASSIGN-BY-GUEST attribute can be set in the VM2000 parameter file in the monitor system (see "Setting the VM2000 parameters in the parameter service").

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 when the VM is initialized (see "Initializing a VM") or with /MODIFY-VM-ATTRIBUTES; the assignment sets required are then assigned to the VM.

> Assignment sets for a VM can be entered in the VM definition with /CREATE- or /MODIFY-VM-DEFINITION. Detailed information can be found in the section "Working with VM definitions".

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 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), the devices can be ascertained 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 [13] and section "SHOW-DEVICE-CONFIGURATION (Display system configuration and availability)").

> **i** 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 (see section "Usage mode of a device"):

- Disk devices and CRD devices are assigned for shared use (usage mode SHARED).
  Exception: FastDPAV alias devices (SU /390) and emulated disk devices (MU) are assigned for exclusive use.
- Other devices are assigned for exclusive use (usage mode EXCLUSIVE).
- Devices that are already assigned exclusively to another VM cannot be assigned implicitly (exception: MTC devices in the KEEP state, see "Implicit release of devices (state KEEP)").

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 may be preserved after `/DETACH-DEVICE`. A device like this takes on the status `KEEP`. Additionally, implicitly assigned MTC devices and emulated disks get the status `KEEP` temporarily during a VM Migration (SU /390).

  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 of the monitor system, the devices assigned implicitly to the monitor VM are released before `SYSTEM READY`. Message `VMS5227` (SU /390) resp. message `VMS4009` with Insert `BY IPL` (SU x86) 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 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 "Explicit device assignment and removal"). 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 `/ADD-VM-DEVICES` with the operand `TYPE=*NONE/*VC`.
- The device (unless it is a disk device or CRD 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 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 "Parallel access volume (PAV, SU /390)").

`/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").

Shared use is said to apply in the following cases:

- The device is assigned to the VM explicitly by `/ADD-VM-DEVICES` with the operand `TYPE=*SD`.
- The disk device (neither FastDPAV alias device nor emulated disk) or CRD 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 "Parallel access volume (PAV, SU /390)").

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.

The usage mode (`SHARED`) cannot be modified for CRD devices.

> The usage mode of a device entered for the VM in its VM definition can be modified with `/MODIFY-VM-DEFINITION`. Detailed information can be found in the section "Working with VM definitions".

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.

x86 On SU x86, 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, the inputs/outputs for the shared device are coordinated.

390 On SU /390, 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.

x86 On SU x86, 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 changs when VM2000 and a guest system are used together.

Guest system operations are marked with "`***`". Asynchronous messages are marked with `%`.

Explicit device assignment

```
/create-vm memory-size=512,vm-name=system2,cpu-quota=30 ————————————————  (1)
VMS3050 VIRTUAL MACHINE (2,SYSTEM2) CREATED
VMS4615 VIRTUAL MACHINE (2,SYSTEM2) ASSIGNED TO CPU POOL '*STDPOOL'
/add-vm-devices (ch,ci,fc1a,fc1b),vm-id=system2 ——————————————————————  (2)
VMS4005 DEVICE 'CH' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'CI' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'FC1A' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'FC1B' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
/start-vm ipl-unit=fc1a,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 devices of the home pubset to VM2. These devices are required by VM2 for startup and thus cannot be assigned implicitly.

(3) VM2 is started.

(4) The guest system on the VM2 has reached `SYSTEM READY`.

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=(CC42,CC43,990f,FC1C,FC1D) ————————————————  (7)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
CC42 HNC
CC43 HNC
990F D3435                    SH(D)  1 MONITOR  VM-ADMIN ————————————— (8)
FC1C D3435
FC1D D3435
```

```
/modify-vm-device-attributes units=(CC42,CC43,990f,fc1c,fc1d),
                             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)
CC42 HNC              SYS2IMPL
CC43 HNC              SYS2IMPL
990F D3435            SYS2IMPL    SH(D)  1 MONITOR  VM-ADMIN
FC1C D3435            SYS2IMPL
FC1D D3435            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=(CC42,CC43,990f,fc1c,fc1d) ———————————————————— (12)
*** NKR0042 'DEVICE     =CC42': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =CC43': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =990F': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =FC1C': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =FC1D': ATTACH ACCEPTED
*** NKR0040 'DEVICE     =CC42' ATTACHED
*** NKR0040 'DEVICE     =CC43' ATTACHED
*** NKR0040 'DEVICE     =FC1C' ATTACHED
*** NKR0040 'DEVICE     =990F' ATTACHED
*** NKR0040 'DEVICE     =FC1D' ATTACHED
*** NKVD010 DISK '4IVL.0' MOUNTED ON DEVICE 'FC1C' ———————————————————— (13)
*** NKVD010 DISK 'WO990F' MOUNTED ON DEVICE '990F'
*** NKVD010 DISK '4IVL.1' MOUNTED ON DEVICE 'FC1D'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%  VMS4005 DEVICE '990F' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
           (BY GUEST) ———————————————————————————————————————————— (14)
%  VMS4005 DEVICE 'CC42' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
%  VMS4005 DEVICE 'CC43' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
%  VMS4005 DEVICE 'FC1C' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
%  VMS4005 DEVICE 'FC1D' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
/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
CC42 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST ————————————— (16)
CC43 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST
990F D3435            SYS2IMPL    SH(I)    ( 2)     MIXED———————————— (17)
                                          1 MONITOR  VM-ADMIN
                                          2 SYSTEM2  GUEST
FC1C D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST ————————————— (18)
FC1D D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST
```

```
/show-vm-attributes information=*all,vm-id=system2 ------------------------ (19)
VM-ID        VM-GROUP(G)  CPU-Q   MAX-U  PROC   MEM-SIZE  MAX-IO
 2 SYSTEM2   *STDPOOL(P)  30.00  100.00  1(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   RUNNING         V20.0B  NO  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    DC    NO    00(RUN),01(SLEEP)
VM-ID        VM-MONJV
  2 SYSTEM2  NO VM-MONJV ASSIGNED
VM-ID        UNITS ------------------------------------------------------- (20)
 2 SYSTEM2   ASSIGNED BY VM-ADMIN
             EXCL:   CH,CI,FC1A,FC1B
             ASSIGNED BY GUEST
             EXCL:   CC42,CC43
             SHARED: 990F,FC1C,FC1D
...
VM-ID        PRIV    DETAILS
 2 SYSTEM2   AG      SYS2IMPL ------------------------------------------- (21)
             AS
*** Guest system operation on VM2:
*** /import-pubset pubset=4ivl,use=*exclusive ------------------------------ (22)
*** DMS035B IMPORT PUBSET TASK WITH TSN 'XAAS' FOR PUBSET WITH PUBSET ID
***         '4IVL' HAS BEEN CREATED AND STARTED
*** SRM5017 USER GROUP MANAGEMENT ON PUBSET '4IVL'INITIALIZED WITH '1'
***         USER GROUP(S)
*** PRO6000 GUARDS CATALOG ':4IVL:$TSOS.SYSCAT.GUARDS' OPENED
*** DMS03E0: SCA (SPEEDCAT) VERSION V19.0A00 RUNNING FOR CATALOG-ID 4IVL
*** DMS03E1: SCA RUNNING FOR CATALOG-ID 4IVL.
***          STATUS OF CATALOG: 00 002664 FILE ENTRIES AND
***          00000002 JOB VARIABLES IN 00000288 CATALOG BLOCKS
*** DMS038F IMPORT PUBSET PROCESSING COMPLETED .
***         PUBSET WITH PUBSET ID '4IVL' IN 'LOCAL-IMPORTED' STATE
*** NBR0740 COMMAND COMPLETED 'IMPORT-PUBSET';
***         (RESULT: SC2=000, SC1=000, MC=DMS038F); DATE: <date>
*** /show-device-configuration unit=(fc1c,fc1d),information=*vm2000 ---- (23)
*** MNEM UN-CLASS UN-TYPE  CONF-STATE   VM-ASSIGN      VM-USAGE ---------- (24)
*** FC1C DVC      D3435    ATTACHED     BY-GUEST       EXCL ------------- (25)
*** FC1D DVC      D3435    ATTACHED     BY-GUEST       EXCL
*** NBR0740 COMMAND COMPLETED 'SHOW-DEVICE-CONFIGURATION'; (RESULT:
*** SC2=000, SC1=000, MC=CMD0001); DATE: <date>
```

---

(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 assign free devices to the VM implicitly (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 state of the devices first.

(8) All the relevant devices are free.
Disk device 990F has already been assigned explicitly (VM-ADMIN) to the monitor VM for shared use with other VMs and therefore has the usage mode SH(D) (shared with direct I/O).

(9) Implicit device assignment is to be permitted for the devices. Therefore, the VM2000 administrator assigns the ASSIGN-BY-GUEST attribute to the relevant devices and assigns them to the assignment set SYS2IMPL.

(10) Now, all the devices concerned have the ASSIGN-BY-GUEST attribute and are assigned to assignment set SYS2IMPL.

(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.

(13) Disks are detected in the guest system on the attached devices.

(14) The VM and VM2000 administrators are notified about the implicit device assignment to VM2 by asynchronous messages.

(15) The VM2000 administrator obtains detailed information on the state 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) Devices which are not disk devices or CRD devices are assigned to the usage mode EXCLUSIVE in the event of implicit device assignment.

(17) Disk device 990F is assigned to the monitor VM explicitly and to VM2 implicitly. The usage mode is therefore SH(I) (shared with indirect I/O).

(18) 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.

(19) This command provides a VM administrator with information on his or her VM.

(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 4IVL is imported in exclusive access mode. The pubset consists of the disk devices FC1C and FC1D.

(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 FC1C and FC1D, 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
CH   SKP2                         EXCL   2 SYSTEM2  VM-ADMIN
CI   SKP2                         EXCL   2 SYSTEM2  VM-ADMIN
CC42 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST
CC43 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST
990F D3435            SYS2IMPL    SH(I)    ( 2)     MIXED
                                          1 MONITOR  VM-ADMIN
                                          2 SYSTEM2  GUEST
FC1A D3435                        EXCL   2 SYSTEM2  VM-ADMIN
FC1B D3435                        EXCL   2 SYSTEM2  VM-ADMIN
FC1C D3435            SYS2IMPL    EXCL   2 SYSTEM2  GUEST ————————————— (27)
FC1D D3435            SYS2IMPL    EXCL   2 SYSTEM2  GUEST
/modify-vm-device-usage units=(fc1c,fc1d),vm-id=system2,usage=*shared — (28)
/show-vm-device-status units=(fc1c,fc1d)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
FC1C D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST  ————————————— (29)
FC1D D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST
```

---

(26) The VM2000 administrator obtains detailed information on the state of the devices assigned to VM2.

(27) For the disk devices, the exclusive importing of the pubset has changed the usage mode from SHARED to EXCLUSIVE, see (25).

(28) Nevertheless, the VM2000 administrator wants to set the usage mode SHARED for these disk devices 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=4ivl ──────────────────────────────────────── (30)
*** DMS035D EXPORT-PUBSET TASK WITH TSN 'XAAV' FOR PUBSET WITH PUBSET ID
***         '4IVL' HAS BEEN CREATED AND STARTED
*** PRO6005 GUARDS CATALOG ':4IVL:$TSOS.SYSCAT.GUARDS' CLOSED
*** DMS03E2 NORMAL TERMINATION OF SCA (CATALOG ID '4IVL')
*** DMS037F EXPORT-PUBSET PROCESSING COMPLETED FOR PUBSET '4IVL'
*** NBR0740 COMMAND COMPLETED 'EXPORT-PUBSET';
***         (RESULT: SC2=000, SC1=000, MC=DMS037F); DATE: <date>
*** /detach-device unit=(fc1c,fc1d) ──────────────────────────────────── (31)
*** NKR0092 'DEVICE     =FC1C': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0092 'DEVICE     =FC1D': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0045 *** 'DEVICE     =FC1C' DETACHED ***
*** NKR0045 *** 'DEVICE     =FC1D' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%  VMS4009 DEVICE 'FC1C' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
           (BY GUEST) ──────────────────────────────────────────────────── (32)
%  VMS4009 DEVICE 'FC1D' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
/show-vm-device-status units=(fc1c,fc1d),information=*all
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
FC1C D3435            SYS2IMPL ──────────────────────────────────────── (33)
FC1D D3435            SYS2IMPL
```

---

(30) Pubset `4IVL` 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 devices have their original status again.

Converting from explicit to implicit device assignment

```
/add-vm-devices units=(fc1c,fc1d),vm-id=system2 ──────────────────────── (34)
VMS4005 DEVICE 'FC1C' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'FC1D' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
/show-vm-device-status units=(fc1c,fc1d),information=*std
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY ──────── (35)
FC1C D3435            SYS2IMPL    EXCL   2 SYSTEM2  VM-ADMIN
```

```
FC1D D3435            SYS2IMPL    EXCL   2 SYSTEM2  VM-ADMIN
***
*** Guest system operation on VM2:
*** /attach-device unit=(fc1c,fc1d) ──────────────────────────────────────── (36)
*** NKR0042 'DEVICE     =FC1C': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =FC1D': ATTACH ACCEPTED
*** NKR0040 'DEVICE     =FC1C' ATTACHED
*** NKR0040 'DEVICE     =FC1D' ATTACHED
*** NKVD010 DISK '4IVL.0' MOUNTED ON DEVICE 'FC1C'
*** NKVD010 DISK '4IVL.1' MOUNTED ON DEVICE 'FC1D'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
/remove-vm-devices units=(fc1c,fc1d),vm-id=system2 ──────────────────────── (37)
VMS4085 DEVICE 'FC1C' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
VMS4085 DEVICE 'FC1D' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
/remove-vm-devices units=(fc1c,fc1d),vm-id=system2,
                   force=*no(wait-for-unassign=*yes) ──────────────────── (38)
VMS4087 DEVICE 'FC1C' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN
        IN GUEST SYSTEM HAS BEEN DONE
VMS4087 DEVICE 'FC1D' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN
        IN GUEST SYSTEM HAS BEEN DONE
/show-vm-device-status units=(fc1c,fc1d),information=*std
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY ──────── (39)
FC1C D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST
FC1D D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST
***
*** Guest system operation on VM2:
*** /detach-device unit=(fc1c,fc1d) ──────────────────────────────────────── (40)
*** %  NKR0092 'DEVICE     =FC1C': /DETACH-DEVICE PENDING ACCEPTED
*** %  NKR0092 'DEVICE     =FC1D': /DETACH-DEVICE PENDING ACCEPTED
*** %  NKR0045 *** 'DEVICE     =FC1C' DETACHED ***
*** %  NKR0045 *** 'DEVICE     =FC1D' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%  VMS4009 DEVICE 'FC1C' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
          (BY GUEST) ──────────────────────────────────────────────────── (41)
%  VMS4009 DEVICE 'FC1D' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
/show-vm-device-status units=(fc1c,fc1d),information=*std
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY ──────── (42)
FC1C D3435            SYS2IMPL
FC1D D3435            SYS2IMPL
```

---

(34) The VM2000 administrator assigns the pubset devices explicitly and exclusively to VM2.

(35) The output of `/SHOW-VM-DEVICE-STATUS` indicates this.

(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).

(39) The output of `/SHOW-VM-DEVICE-STATUS` indicates that the devices are now assigned implicitly to the VM. At the same time, the usage mode has also changed (`SH(D)`), see also (18).

(40) The devices are detached in the guest system and consequently released implicitly.

(41) VM and VM2000 administrators are informed of the implicit device release by asynchronous messages.

(42) The devices are free once 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
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,SYSTEM2),
           CODE 80FFFFFF, CPU 01
%  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
%  VMS4009 DEVICE 'CC42' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) ———— (44)
%  VMS4009 DEVICE 'CC43' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
%  VMS4009 DEVICE '990F' 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)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   DOWN            V20.0B  NO  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    -     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:   CH,CI,FC1A,FC1B ———————————————————————————— (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 'CH' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'CI' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'FC1A' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'FC1B' 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".

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 "Pubset reconfiguration") are then not available.

Pubres device (system disk of a pubset)

In BS2000 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 "Explicit pubset assignment and removal") and `PUBRES-FOR-ATTRIBUTES` (see "Implicit pubset assignment and release") 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

390 On SU /390, the devices of the home pubset are automatically assigned explicitly to the monitor VM at startup of the monitor system.

x86 On SU x86, the devices of the home pubset must be assigned explicitly via the SE manager before the monitor system is started up, see "Configuration using the SE Manager (SU x86)".

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 if displayed implicitly via the use of the pubset devices in the VM2000 accounting record for devices (see "Accounting record for devices").

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.

> Pubsets for a VM can be entered in its VM definition with /CREATE- or /MODIFY-VM-DEFINITION. The pubset is represented in the (inactive) VM definition only by the pubres device (suffix (PR) in /SHOW-VM-DEFINITION). Only when the VM definition is activated are the other pubset devices also determined and entered in the VM definition (suffix (PD) in /SHOW-VM-DEFINITION). When the VM is terminated, the pubset devices (except for the pubres device) are removed from the VM definition. Detailed information can be found in the section "Working with VM definitions".

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 bay 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 "Assignment sets, implicit device assignment and release".

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 when the VM is initialized (see "Initializing a VM") or with /MODIFY-VM-ATTRIBUTES; the assignment sets required are then assigned to the VM.

> ¡ Assignment sets for a VM can be entered in the VM definition with /CREATE- or /MODIFY-VM-DEFINITION. Detailed information can be found in the section "Working with VM definitions".

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), the devices can be ascertained which a guest system may assign itself implicitly.

The BS2000 command /SHOW-DEVICE-CONFIGURATION UNIT=*PUBSET-DEVICES(...), INFORMATION=*VM2000 provides information on the options for implicit pubset assignment to the local VM in the guest system, see the “Commands” manual [13] and section "SHOW-DEVICE-CONFIGURATION (Display system configuration and availability)".

> 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 pubset 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 "Explicit pubset assignment and removal". 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".

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 `/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 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".

Shared use is said to apply in the following cases:

- The pubset is assigned to the VM explicitly by `/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.

> The usage mode of a pubset entered for the VM in its VM definition can be modified with `/MODIFY-VM-DEFINITION`. Detailed information can be found in the section "Working with VM definitions".

For information on input/output handling for the devices of shared pubsets, please refer to the paragraph “Input /output handling for shared devices” in the section "Usage mode of a device".

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 during pubset operation (dynamic pubset reconfiguration). Basic information on this is provided in the section “Dynamic pubset reconfiguration” in the “Introduction to System Administration” [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 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, 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 paragraph "Adjustments by VM2000". In special cases, manual adjustments by the VM2000 administrator may be required, see paragraph "Manual adjustments by the VM2000 administrator".

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 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 Unit for all the guest systems involved, provided at least one of these guest systems is running.
- The systems support of a guest system 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 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 pubres device has assumed the `PUBRES-FOR-ASSIGNMENT` attribute for the VM, and the new pubset device is not assigned to the VM explicitly, 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 "MODIFY-VM-DEVICE-ATTRIBUTES (Modify device attributes)") 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 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 on the Server Unit for all the guest systems involved.

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 state a volume is removed from the pubset by a system on another Server Unit using /MODIFY-PUBSET-PROCESSING.
- If the home pubset is assigned explicitly to a VM in the DOWN state and in this state volumes are added to or removed from the home pubset by another system using /MODIFY-PUBSET-PROCESSING.

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 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 "Assigning and removing pubsets of a VM".

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 "Assigning and removing pubsets of a VM".

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 "Pubset reconfiguration".

Mirror disks in the local disk storage system

Clone units 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 [20].

The following procedure applies for clone units in VM2000:

- When a pubset with clone units is operated, the required VM2000 function is executed on the pubset’s original units when the `*PUBSET-DEVICES` operand is specified, provided the original 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 can also be specified. The required VM2000 function is then executed on the set of the pubset’s clone units which belongs to the specified clone unit of the pubres device. In this case the software product SHC-OSD is also required in the monitor system.

> **i** 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.

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 "Assigning and removing pubsets of a VM".

> The statements above apply generally for local replication functions of external disk storage systems, provided these functions are supported by BS2000 and SHC-OSD.

Mirror disks in remote disk storage systems

When the REC function (Remote Equivalent Copy, ETERNUS DX) is used, the data of a local disk storage system is mirrored on **remote** disk storage systems, see the "SHC-OSD" manual [20].

For such a pubset, 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 registered in the MRS catalog.

Only if VM2000 can access the target units in the remote disk storage system, a target unit can 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 "Assigning and removing pubsets of a VM".

> The statements above apply generally for remote replication functions of external disk storage systems, provided these functions are supported by BS2000 and SHC-OSD.

6.4.2.8 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", but it utilizes simplified device selection in the command by specifying the pubset.

Guest system operations are marked with "`***`". Asynchronous messages are marked with `%`.

Explicit pubset assignment

```
/create-vm memory-size=512,vm-name=system2,cpu-quota=30 ————————————  (1)
VMS3050 VIRTUAL MACHINE (2,SYSTEM2) CREATED
VMS4615 VIRTUAL MACHINE (2,SYSTEM2) ASSIGNED TO CPU POOL '*STDPOOL'
/add-vm-devices units=(ch,ci),vm-id=system2 ———————————————————————  (2)
VMS4005 DEVICE 'CH' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'CI' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
/add-vm-devices units=*pubset-devices(pubset=4ivk),vm-id=system2
VMS4005 DEVICE 'FC1A' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'FC1B' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4220 PUBSET '4IVK' WITH PUBRES DEVICE 'FC1A' ASSIGNED TO VM (2,SYSTEM2)
/start-vm ipl-unit=fc1a,vm-id=system2 ————————————————————————————  (3)
VMS2023 CPU 00 OF VIRTUAL MACHINE (2,SYSTEM2) STARTED
VMS2023 CPU 01 OF VIRTUAL MACHINE (2,SYSTEM2) STARTED
***
*** 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 devices of the home pubset to VM2. These devices are required by VM2 for startup and thus cannot be assigned implicitly. Pubset `4IVK` is registered in the monitor system's MRS catalog and is consequently detected by VM2000.

(3) VM2 is started.

(4) The guest system on the VM2 has reached `SYSTEM READY`.

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=4ivl) ---------------  (7)
%  NKR0042 'DEVICE     =FC1C': ATTACH ACCEPTED
%  NKR0040 'DEVICE     =FC1C' ATTACHED
%  NKVD010 DISK '4IVL.0' MOUNTED ON DEVICE 'FC1C'
%  NKR0045 *** 'DEVICE     =FC1C' DETACHED ***
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
FC1C D3435
FC1D D3435
/show-vm-device-status units=(CC42,CC43,990f)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
CC42 HNC
CC43 HNC
990F D3435                        SH(D)  1 MONITOR  VM-ADMIN ------------  (8)
/modify-vm-device-attributes units=*pubset-devices(pubset=4ivl),
                 assign-by-guest=*yes(set-name=sys2impl) ---------------  (9)
VMS4240 DEVICE ATTRIBUTES OF PUBSET '4IVL' WITH PUBRES DEVICE 'FC1C' MODIFIED
/modify-vm-device-attributes units=(CC42,CC43,990f),
                           assign-by-guest=*yes(set-name=sys2impl)
/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)
CC42 HNC              SYS2IMPL
CC43 HNC              SYS2IMPL
990F D3435            SYS2IMPL    SH(D)  1 MONITOR  VM-ADMIN
FC1C D3435            SYS2IMPL(P)
FC1D D3435            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=*pubset-devices(pubset=4ivl) -------------------  (12)
*** %  NKR0042 'DEVICE     =FC1C': ATTACH ACCEPTED
*** %  NKR0040 'DEVICE     =FC1C' ATTACHED
*** %  NKR0042 'DEVICE     =FC1D': ATTACH ACCEPTED
*** %  NKVD010 DISK '4IVL.0' MOUNTED ON DEVICE 'FC1C' ------------------  (13)
*** %  NKR0040 'DEVICE     =FC1D' ATTACHED
*** %  NKVD010 DISK '4IVL.1' MOUNTED ON DEVICE 'FC1D'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
*** /attach-device unit=(CC42,CC43,990f)
*** NKR0042 'DEVICE     =CC42': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =CC43': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =990F': ATTACH ACCEPTED
*** NKR0040 'DEVICE     =CC42' ATTACHED
*** NKR0040 'DEVICE     =CC43' ATTACHED
*** NKR0040 'DEVICE     =990F' ATTACHED
```

```
*** NKVD010 DISK 'WO990F' MOUNTED ON DEVICE '990F'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%  VMS4005 DEVICE 'FC1C' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)   (BY GUEST)
%  VMS4005 DEVICE 'FC1D' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)   (BY GUEST)
%  VMS4005 DEVICE 'CC42' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
          (BY GUEST) ──────────────────────────────────────────────── (14)
%  VMS4005 DEVICE 'CC43' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)   (BY GUEST)
%  VMS4005 DEVICE '990F' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)   (BY GUEST)
/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
CC42 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST ──────────── (16)
CC43 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST
990F D3435            SYS2IMPL    SH(I)    ( 2)     MIXED ──────────── (17)
                                          1 MONITOR  VM-ADMIN
                                          2 SYSTEM2  GUEST
FC1C D3435            SYS2IMPL(P) SH(D)  2 SYSTEM2  GUEST ──────────── (18)
FC1D D3435            SYS2IMPL    SH(D)  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-IO
 2 SYSTEM2  *STDPOOL(P)  30.00  100.00  1(+1)       512  (100)
VM-ID       STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2  RUNNING         V20.0B  NO  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   DC    NO    00(RUN),01(SLEEP)
VM-ID       VM-MONJV
 2 SYSTEM2  NO VM-MONJV ASSIGNED
VM-ID       UNITS ──────────────────────────────────────────────────── (20)
 2 SYSTEM2  ASSIGNED BY VM-ADMIN
            EXCL:   CH,CI,FC1A,FC1B
            ASSIGNED BY GUEST
            EXCL:   CC42,CC43
            SHARED: 990F,FC1C,FC1D
...
VM-ID       PRIV    DETAILS
 2 SYSTEM2  AG      SYS2IMPL ──────────────────────────────────────── (21)
            AS
*** Guest system operation on VM2:
*** /import-pubset pubset=4ivl,use=*exclusive ──────────────────────── (22)
*** DMS035B IMPORT PUBSET TASK WITH TSN 'XAAU' FOR PUBSET WITH PUBSET ID
***         '4IVL' HAS BEEN CREATED AND STARTED
*** SRM5017 USER GROUP MANAGEMENT ON PUBSET '4IVL' INITIALIZED WITH '1'
```

```
***         USER GROUP(S)
*** PRO6000 GUARDS CATALOG ':4IVL:$TSOS.SYSCAT.GUARDS' OPENED
*** DMS03E0: SCA (SPEEDCAT) VERSION V19.0A00 RUNNING FOR CATALOG-ID 4IVL
*** DMS03E1: SCA RUNNING FOR CATALOG-ID 4IVL.
***          STATUS OF CATALOG: 00 002664 FILE ENTRIES AND
***          00000002 JOB VARIABLES IN 00000288 CATALOG BLOCKS
*** DMS038F IMPORT PUBSET PROCESSING COMPLETED.
***         PUBSET WITH PUBSET ID '4IVL' IN 'LOCAL-IMPORTED' STATE
*** NBR0740 COMMAND COMPLETED 'IMPORT-PUBSET';
***         (RESULT: SC2=000, SC1=000, MC=DMS038F); DATE: <date>
*** /show-device-config unit=*pubset-devices(pubset=4ivl),info=*vm2000   (23)
*** MNEM UN-CLASS UN-TYPE  CONF-STATE   VM-ASSIGN      VM-USAGE ———————— (24)
*** FC1C DVC      D3435    ATTACHED     BY-GUEST       EXCL ———————————— (25)
*** FC1D DVC      D3435    ATTACHED     BY-GUEST       EXCL
*** NBR0740 COMMAND COMPLETED 'SHOW-DEVICE-CONFIGURATION';
*** (RESULT:SC2=000, SC1=000, MC=CMD0001); DATE: <date>
```

---

(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 assign free devices to the VM implicitly (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 state of the devices first. The pubres device FC1C 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 device 990F has already been assigned explicitly (VM-ADMIN) to the monitor VM for shared use with other VMs and therefore has the usage mode SH(D) (shared with direct I/O).

(9) Implicit device assignment is to be permitted for the devices. Therefore, the VM2000 administrator specifies the ASSIGN-BY-GUEST attribute for the pubset and the devices concerned and assigns them to assignment set SYS2IMPL.

(10) Now, all the devices concerned have the ASSIGN-BY-GUEST attribute and are assigned to assignment set SYS2IMPL. The PUBRES-FOR-ATTRIBUTES attribute is displayed for device FC1C.

(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 pubset. The devices of the pubset are assigned to VM2 implicitly.

(13) Disks are detected in the guest system on the attached disk devices.

(14) The VM and VM2000 administrators are notified about the implicit device assignment to VM2 by asynchronous messages.

(15) The VM2000 administrator obtains detailed information on the state 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) Devices which are not disk devices or CRD devices are assigned to the usage mode EXCLUSIVE in the event of implicit device assignment.

(17) Disk device 990F is assigned to the monitor VM explicitly and to VM2 implicitly. The usage mode is therefore SH(I) (shared with indirect I/O).

(18) 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.

(19) This command provides a VM administrator with information on his or her VM.

(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 4IVL is imported in exclusive access mode. The pubset consists of the disk devices FC1C and FC1D.

(23) The guest system operator obtains information on the VM2000 state 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 FC1C and FC1D, 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
CH   SKP2                         EXCL   2 SYSTEM2  VM-ADMIN
CI   SKP2                         EXCL   2 SYSTEM2  VM-ADMIN
CC42 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST
CC43 HNC              SYS2IMPL    EXCL   2 SYSTEM2  GUEST
990F D3435            SYS2IMPL    SH(I)    ( 2)     MIXED
                                           1 MONITOR  VM-ADMIN
                                           2 SYSTEM2  GUEST
FC1A D3435                        EXCL   2 SYSTEM2  VM-ADMIN(P)
FC1B D3435                        EXCL   2 SYSTEM2  VM-ADMIN
FC1C D3435            SYS2IMPL(P) EXCL   2 SYSTEM2  GUEST ————————————— (27)
FC1D D3435            SYS2IMPL    EXCL   2 SYSTEM2  GUEST

/modify-vm-device-usage vm-id=system2,usage=*shared,
                        units=*pubset-devices(pubset=4ivl) ———————————— (28)
VMS4230 DEVICE USAGE OF PUBSET '4IVL' WITH PUBRES DEVICE 'FC1C' MODIFIED

/show-vm-device-status units=*pubset-devices(pubset=4ivl)

UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY
FC1C D3435            SYS2IMPL(P) SH(D)  2 SYSTEM2  GUEST ————————————— (29)
FC1D D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST
```

---

(26) The VM2000 administrator obtains detailed information on the state of the devices assigned to VM2.

(27) For the disk devices FC1C and FC1D, 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 disk 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=4ivl ──────────────────────────────────────── (30)
*** DMS035D EXPORT-PUBSET TASK WITH TSN 'XAAX'FOR PUBSET WITH PUBSET ID
            '4IVL' HAS BEEN CREATED AND STARTED
*** PRO6005 GUARDS CATALOG ':4IVL:$TSOS.SYSCAT.GUARDS' CLOSED
*** DMS03E2 NORMAL TERMINATION OF SCA (CATALOG ID '4IVL')
*** DMS037F EXPORT-PUBSET PROCESSING COMPLETED FOR PUBSET '4IVL'
*** NBR0740 COMMAND COMPLETED 'EXPORT-PUBSET';
***         (RESULT: SC2=000, SC1=000, MC=DMS037F); DATE: <date>
*** /detach-device unit=*pubset-devices(pubset=4ivl) ──────────────────── (31)
*** NKR0092 'DEVICE     =FC1C': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0092 'DEVICE     =FC1D': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0045 *** 'DEVICE     =FC1C' DETACHED ***
*** NKR0045 *** 'DEVICE     =FC1D' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%  VMS4009 DEVICE 'FC1C' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
           (BY GUEST) ──────────────────────────────────────────────────── (32)
%  VMS4009 DEVICE 'FC1D' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
/show-vm-device-status units=*pubset-devices(pubset=4ivl),information=*all
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY ──────── (33)
FC1C D3435            SYS2IMPL(P)
FC1D D3435            SYS2IMPL
```

---

(30) Pubset `4IVL` 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 devices have their original state again.

Converting from explicit to implicit pubset assignment

```
/add-vm-devices units=*pubset-devices(pubset=4ivl),vm-id=system2 ──────── (34)
VMS4005 DEVICE 'FC1C' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4005 DEVICE 'FC1D' ASSIGNED TO VIRTUAL MACHINE (2,SYSTEM2)
VMS4220 PUBSET '4IVL' WITH PUBRES DEVICE 'FC1C' ASSIGNED TO VM (2,SYSTEM2)
/show-vm-device-status information=*std,units=*pubset-devices(pubset=4ivl)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY ──────── (35)
```

```
FC1C D3435            SYS2IMPL(P) EXCL   2 SYSTEM2  VM-ADMIN(P)
FC1D D3435            SYS2IMPL    EXCL   2 SYSTEM2  VM-ADMIN
*** Guest system operation on VM2:
*** /attach-device unit=*pubset-devices(pubset=4ivl) ———————————— (36)
*** NKR0042 'DEVICE     =FC1C': ATTACH ACCEPTED
*** NKR0040 'DEVICE     =FC1C' ATTACHED
*** NKR0042 'DEVICE     =FC1D': ATTACH ACCEPTED
*** NKVD010 DISK '4IVL.0' MOUNTED ON DEVICE 'FC1C'
*** NKR0040 'DEVICE     =FC1D' ATTACHED
*** NKVD010 DISK '4IVL.1' MOUNTED ON DEVICE 'FC1D'
*** NBR0740 COMMAND COMPLETED 'ATTACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
/remove-vm-devices vm-id=system2,units=*pubset-devices(pubset=4ivl) —— (37)
VMS4085 DEVICE 'FC1C' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
VMS4085 DEVICE 'FC1D' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
VMS4223 VM2000 COMMAND /REMOVE-VM-DEVICES FOR PUBSET SPECIFIED BY PUBRES
        DEVICE OR CATID '4IVL' REJECTED
/remove-vm-devices units=*pubset-devices(pubset=4ivl),vm-id=system2,
                   force=*no(wait-for-unassign=*yes) ———————————— (38)
VMS4087 DEVICE 'FC1C' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN
        GUEST SYSTEM HAS BEEN DONE
VMS4087 DEVICE 'FC1D' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN
        GUEST SYSTEM HAS BEEN DONE
VMS4235 SOME DEVICES OF PUBSET '4IVL' WITH PUBRES DEVICE 'FC1C' REMAIN
        ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN GUEST SYSTEM HAS BEEN DONE
/show-vm-device-status information=*std,
                       units=*pubset-devices(pubset=4ivl)
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY ———————— (39)
FC1C D3435            SYS2IMPL(P) SH(D)  2 SYSTEM2  GUEST
FC1D D3435            SYS2IMPL    SH(D)  2 SYSTEM2  GUEST
***
*** Guest system operation on VM2:
*** /detach-device unit=*pubset-devices(pubset=4ivl) ———————————— (40)
*** NKR0092 'DEVICE     =FC1C': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0092 'DEVICE     =FC1D': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0045 *** 'DEVICE     =FC1C' DETACHED ***
*** NKR0045 *** 'DEVICE     =FC1D' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
***         (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
***
%  VMS4009 DEVICE 'FC1C' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
           (BY GUEST) ———————————————————————————————————————— (41)
%  VMS4009 DEVICE 'FC1D' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) (BY GUEST)
/show-vm-device-status units=*pubset-devices(pubset=4ivl),information=*std
UNIT DEV-TYPE PAV   S ASSIGN-SET  USAGE VM-ID/(#VM) ASSIGNED-BY ———————— (42)
FC1C D3435            SYS2IMPL(P)
FC1D D3435            SYS2IMPL
```

---

(34) The VM2000 administrator assigns the pubset devices explicitly and exclusively to VM2.

(35) The output of `/SHOW-VM-DEVICE-STATUS` indicates this. For the device `FC1C` 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).

(39) The output of `/SHOW-VM-DEVICE-STATUS` indicates that the devices are 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 devices are detached in the guest system and consequently released implicitly.

(41) VM and VM2000 administrators are informed of the implicit device release by asynchronous messages.

(42) The devices are once again free.

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
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,SYSTEM2),
           CODE 80FFFFFF, CPU 01
%  VMS2033 'EXC0557 SHUTDOWN PROCESSING COMPLETED' FROM VM (2,SYSTEM2)
%  VMS2051 GUEST SYSTEM ON VM (2,SYSTEM2) DOWN; REASON: SHUTDOWN
%  VMS2037 NON-INTERRUPTIBLE WAIT STATE FOR VIRTUAL MACHINE (2,SYSTEM2),
           CODE 80FFFFFF, CPU 00
%  VMS4009 DEVICE 'CC42' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2) ——— (44)
%  VMS4009 DEVICE 'CC43' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
%  VMS4009 DEVICE '990F' 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)       512  (100)
VM-ID         STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2    DOWN            V20.0B  NO  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     -     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:   CH,CI,FC1A,FC1B ———————————————————————————— (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 'CH' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'CI' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'FC1A' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'FC1B' 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

Configuration Raw Device (CRD, SU /390)

Special system disks (Configuration Raw Devices (CRDs)) are used for storing the VM definitions (see "Working with VM definitions").

For high-availability functions in an SU cluster, the content of these system disks is also duplicated on disks in an external disk storage system. These external disks can be then announced to all BS2000 systems involved (on SU /390) by means of the hardware generation and made accessible to them over Fibre Channel. In this manual the term Configuration Raw Device (CRD) only refers to these external disks.

CRD devices are always assigned implicitly to a VM, irrespective of the `ASSIGN-BY-GUEST` setting, by the guest system (usage mode `SHARED`), see "Assignment sets, implicit device assignment and release". The VM and CRD device do **not** need to be prepared for the implicit device assignment (i.e. no privilege or attribute `ASSIGN-BY-GUEST` for the VM or device). Explicit assignment of CRD devices is rejected.

CRD devices have the device type `HA-CRD`, visible, for example, in `/SHOW-VM-DEVICE-STATUS INF=*STD/*ALL` in the `DEV-TYPE` column or in the S variable `TYPE`.

Virtual console

It is advisable to operate guest systems via KVP consoles or logical consoles, see "Operating the guest systems using BS2000 consoles".

The operating strategy of VM2000 on SU /390 also provides for an emulation of the guest system console by the VM2000 hypervisor, see section "Operating the guest systems (VC dialog via $VMCONS)".

> Virtual consoles must be assigned to the VM explicitly (`TYPE=*VC`). The attribute `ASSIGN-BY-GUEST=*YES` cannot be set for virtual consoles.

x86 On SU x86, virtual consoles (as generated devices) are not available. Nevertheless, the VC dialog via $VMCONS is possible.

Dummy tape devices

Dummy tape devices defined by systems support are used as backup devices in tape device management in BS2000. 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) 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 "BS2000 devices connected to SU x86".

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 during a hardware generation of the Server Unit (using IOGEN (SU /390) or in X2000 (SU x86)).

VM2000 detects snap units in ETERNUS DX disk storage systems, providing they have been configured there as snap volumes.

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.
> ETERNUS DX disk storage systems also offer snap functionality without dedicated configured snap units. Also in this case, the implicit assignment of Snapsets described below can still be used in VM2000.

Guest systems 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. Snap units are attached automatically when the pubset is imported and, in VM2000, implicitly assigned to the VM.

  In 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 when it is initialized. The privilege is displayed with `AS` in `/SHOW-VM-ATTRIBUTES/-RESOURCES INFORMATION=*PRIVILEGES`.

  > `/CREATE-` or `/MODIFY-VM-DEFINITION` enables the `AUTO-SNAP-ASSIGNMENT` privilege to be entered for a VM in the VM definition. Detailed information can be found in the section "Working with VM definitions".

- 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).

  In 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.

Emulated disk devices on a Management Unit (MU, SU /390)

It is possible to generate emulated disks for SU /390 on an MU (as with emulated local disks on SU x86). These disks are used for an initial BS2000 installation or for a BS2000 emergency system. VM2000 detects these special disk devices and prevents their shared use (usage mode `SHARED`). During a VM Migration, BS2000 automatically detaches them in the guest system (like tape devices and alias devices).

6.4.4 Shared disk devices

VM2000 supports SPVS mode (SPVS=Shared Pubset) of disks.

When the software product HIPLEX MSCF and a corresponding configuration of the BS2000 servers are used, simultaneous access to a shareable pubset is possible via several BS2000 servers (see the “HIPLEX MSCF” manual [10).

Depending on whether the virtual machines wishing to share disks are set up on the same Server Unit or are distributed across several BS2000 servers in a multiprocessor network, the VM2000 administrator must take different measures which are described on the pages below.

In a multiprocessor network, not only Server Units with or without VM2000 operation can execute, but also other BS2000 servers with or without VM2000 operation which wish to work with shareable disks.

The term **BS2000 server** in this section refers to a BS2000 system irrespective of whether it runs on a Server Unit /390 or x86 (on the same SE server or on different SE servers).

At most 16 BS2000 servers can work simultaneously with a shared pubset.

Input/output handling for shared disks

See "Usage mode of a device".

Sharing of disks by several VMs on one Server Unit

The shared use of a disk device by different VMs on one Server Unit is initiated by implicit or explicit device assignment to the VM in `SHARED` usage mode, see "Usage mode of a device" (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 disk devices `D1`, `D2` and `D3` for SPVS mode.

Action to be taken by the VM2000 administrator:

Disk devices `D1` through `D3` must be assigned to each VM as shareable, e.g. with the commands:

```
/ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMx,TYPE=*SD
```

```
/ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMy,TYPE=*SD
```

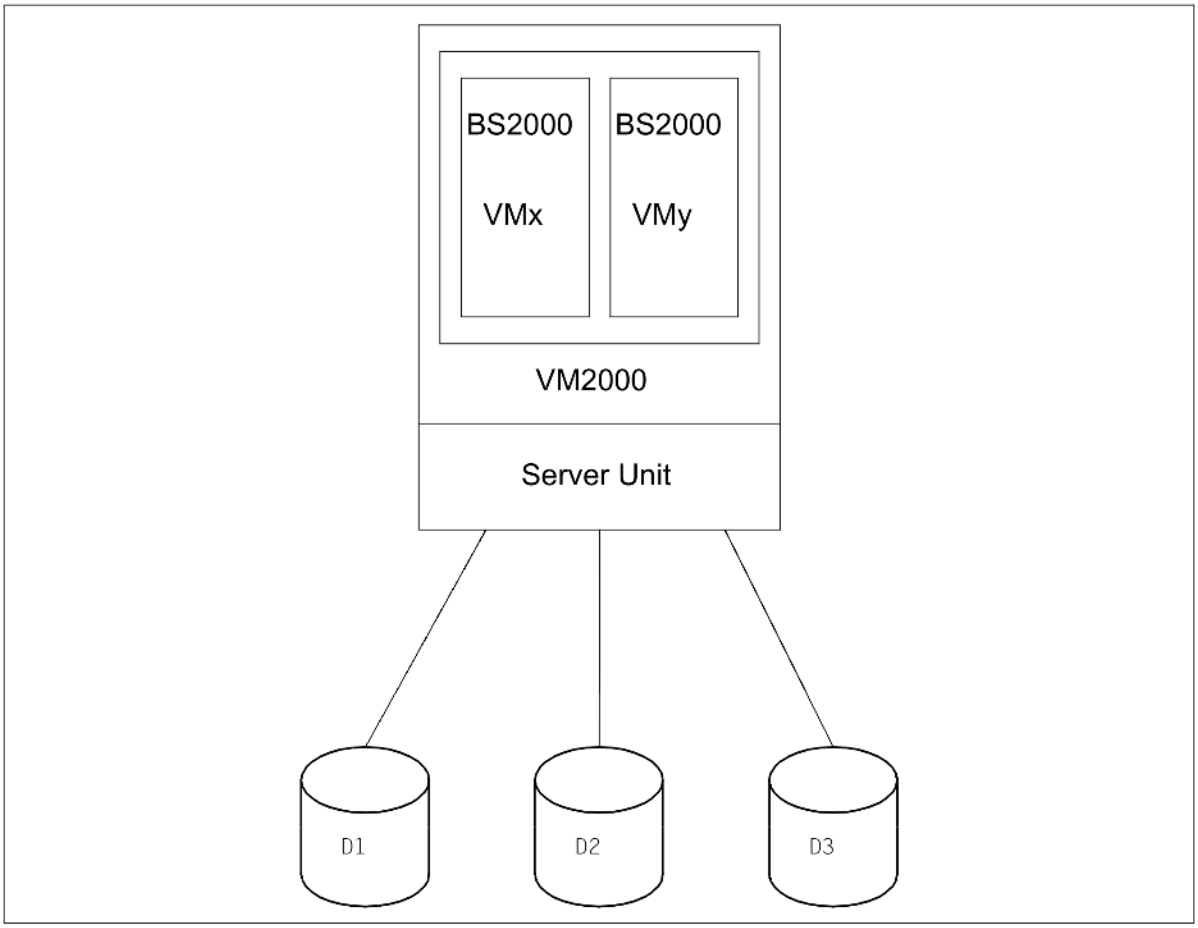


Figure 20: Sharing of disks by several VMs on one Server Unit

Sharing of disks in a VM2000 multiprocessor network

In a multiprocessor environment, several BS2000 servers with VM2000 are operated in parallel. Each BS2000 system can access shared peripheral devices. The BS2000 systems run as guest systems under VM2000.

The following configurations may be possible:

1. Desired configuration:

   On BS2000 server 1 only VMm and on BS2000 server 2 only VMy wish to use disk devices `D1`, `D2` and `D3` for SPVS mode (see Figure 21).

   Action to be taken by the VM2000 administrator:

   Disk devices `D1` through `D3` should be assigned to VMm and VMy **exclusively** with the command:

   ```
   /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMm,TYPE=*NONE
   /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMy,TYPE=*NONE
   ```

2. Desired configuration:

   On BS2000 server 1 only VMm and on BS2000 server 2 both VMs wish to use disk devices `D1`, `D2` and `D3` for SPVS mode (see Figure 21).

Action to be taken by the VM2000 administrator:

- Disk devices `D1` through `D3` should be assigned to VMm **exclusively** with the command:

  ```
  /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMm,TYPE=*NONE
  ```

- Disk devices `D1` through `D3` must be assigned to both VMs (VMx and VMy) as **shareable**, e.g. with the command:

  ```
  /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMx,TYPE=*SD
  /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMy,TYPE=*SD
  ```

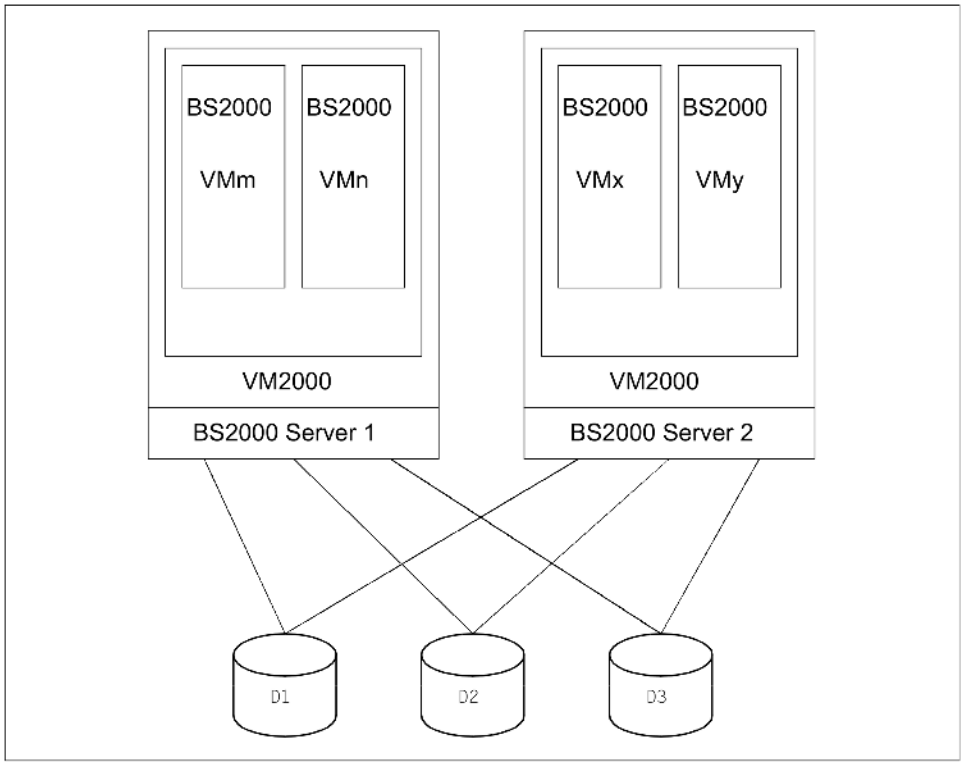


Figure 21: Sharing of disks by VMs on different BS2000 servers

Sharing of disks in a mixed multiprocessor network

This section applies to configurations that meet the following **general** criteria:

- SPVS mode between several BS2000 systems on different servers
- At least one of the BS2000 systems active in SPVS mode is running as a guest system under VM2000
- At least one of the BS2000 systems active in SPVS mode is running in native mode

Example of a mixed multiprocessor network

The Figure 22 shows an example of a configuration of the Fibre Channel peripherals in which, in addition to a native-mode BS2000 system, two other BS2000 servers (on each of which several BS2000 systems are running as guest systems under VM2000) access shareable disk devices.

Desired configuration:

On BS2000 server 1 only VMm and on BS2000 server 2 both VMs wish to use disk devices `D1`, `D2` and `D3` for SPVS mode. On BS2000 server 3 a BS2000 system is operated without VM2000, and, together with the VMs on the other BS2000 servers, it wishes to use the disks for SPVS mode.

Action to be taken by the VM2000 administrator:

- Disk devices `D1` through `D3` should be assigned to VMm on BS2000 server 1 **exclusively** with the command:

  ```
  /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMm,TYPE=*NONE
  ```

- Disk devices `D1` through `D3` are to be assigned to both VMs (VMx and VMy on BS2000 server 2) as **shareable**, e.g. with the commands:

  ```
  /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMx,TYPE=*SD
  /ADD-VM-DEVICES UNITS=(D1,D2,D3),VM-ID=VMy,TYPE=*SD
  ```

- BS2000 on server 3 acts like a BS2000 in the native network (see also the “HIPLEX MSCF” manual [10]).

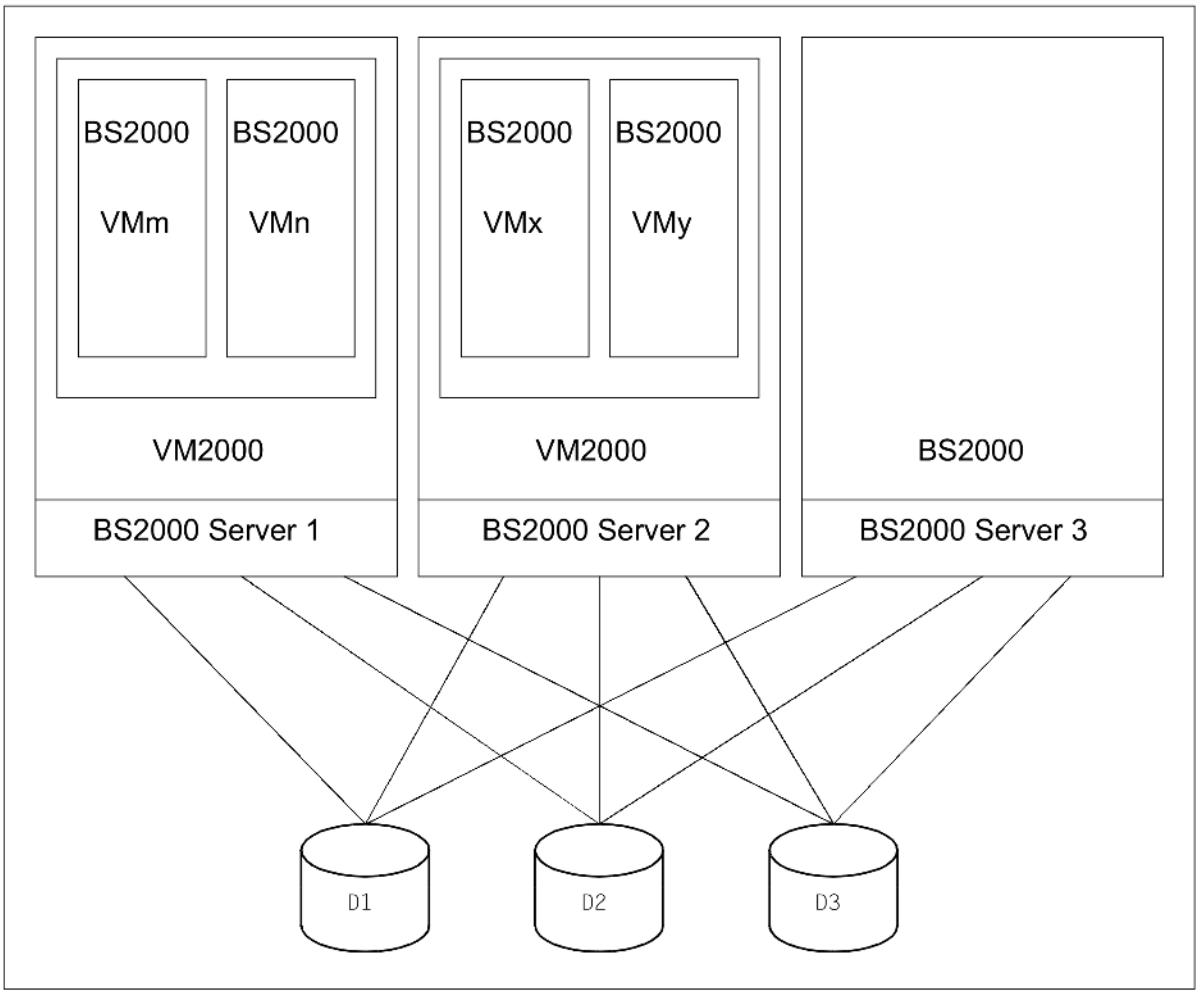


Figure 22: Sharing of channel disks in the mixed multiprocessor network

6.4.5 Parallel access volume (PAV, SU /390)

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.

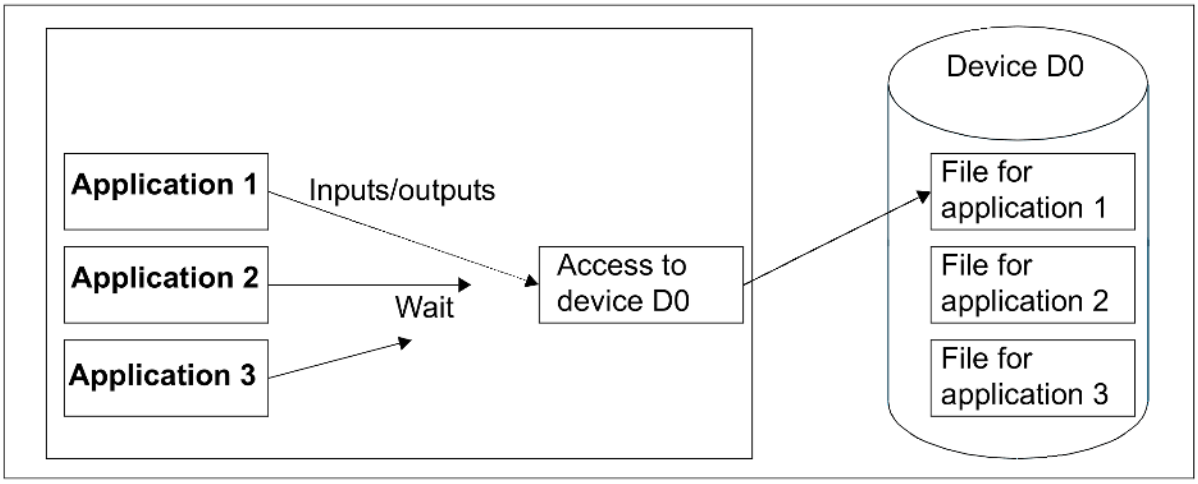


Figure 23: Applications with input/output load on one device (serialized)

For all disk storage systems operated via Fibre Channel, BS2000 offers the function Parallel Access Volume (PAV) which enables multiple I/Os to a disk device (a logical volume) to be performed in parallel.

x86 PAV devices are not available on SU x86. The inputs/outputs there are automatically parallelized by X2000.

A PAV device consists of a real **base device** (BASE) with up to seven **alias devices** (ALIAS).

Further information on PAV is provided in the “Introduction to System Administration” [2].

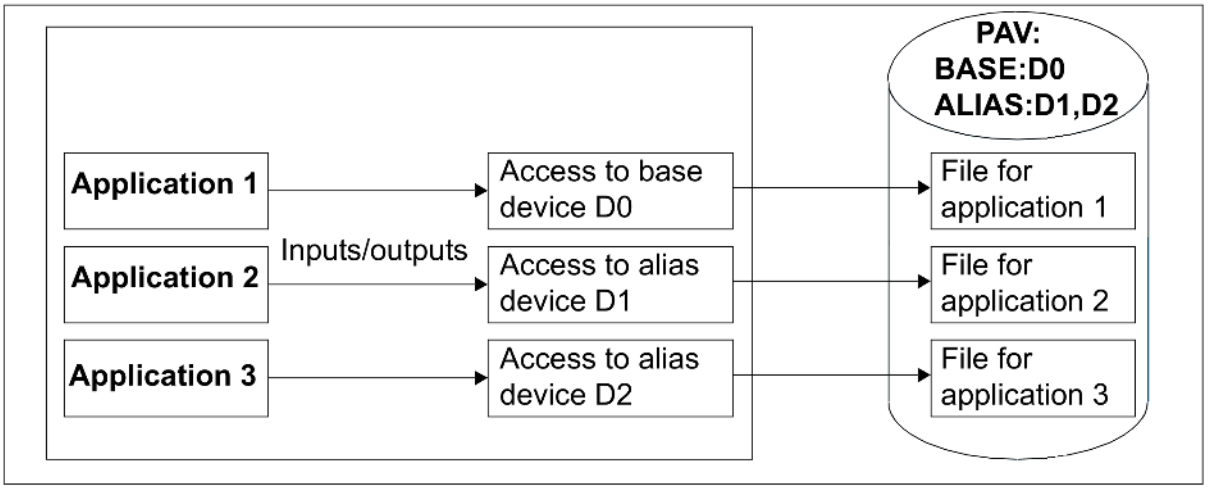


Figure 24: Applications with input/output load on one device (parallel access volume)

(Static) PAV in native operation

PAV can be used on the channel for disk devices of the type D3435 in FBA format.

Here the PAVs are set up only when the I/O 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 I/O configuration, see "Dynamic PAV in native operation".

At hardware generation (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 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 controls execution via the base or alias device.

The BS2000 command `/SHOW-DEVICE-CONFIGURATION INFORMATION=*PAV` provides information on PAVs.

(Static) PAV in VM2000 operation

PAVs can be used in VM2000 for the monitor system and the guest systems.

The base device can be assigned to a VM explicitly or implicitly (see "Assigning and removing devices of a VM").

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 "Usage mode of a device".

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 "Assignment sets, implicit device assignment and release"). Alias devices are released automatically with the base device. Explicit removal of alias devices is rejected.

You can obtain information on PAVs in 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 distribution of I/Os for VM2000 shared disks as possible, the operating system 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:
`/MODIFY-IO-UNIT UNIT=*DEVICE(NAME=...,STATE=*PAV-PREFERRED-DEVICE(...))`
(see "MODIFY-IO-UNIT (Modify configuration properties of an I/O unit)").

Dynamic PAV (DPAV) in native operation

Static PAV, as described above, requires careful planning of future device utilization. When generation in BS2000 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 "Use of IORM in VM2000 operation" 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 on the SU /390 for disk devices on the channel in both modes, native and VM2000.

Dynamic PAV (DPAV) in VM2000 operation

DPAV in 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. 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 assigned to its own VM implicitly - in the guest systems in which the base device is attached.

> For Server Units that support FastDPAV, it is no longer recommended to use DPAV.

Fast Dynamic PAV (FastDPAV) in native operation

When a bottleneck occurs for a PAV device in a guest system using DPAV, the above described actions (especially in VM2000 operation) need certain time, in which the load for PAV devices might have changed again. In order to bybass this shortcoming, an optimized DPAV ("FastDPAV") is offered for Server Units /390 that supports a modification of the logical unit number (LUN) for alias devices when starting an I/O (SU710 and later).

When using FastDPAV, a pool of FastDPAV alias devices is generated (with device typ D3435-FP, internally A6) for a set of logical volumes with identical channel paths without a permanent assignment to one of these logical volumes. For a FastDPAV alias device, the LUN is only defined/modified when an I/O is started.

Advantages of FastDPAV compared to DPAV:

- FastDPAV does not require IORM for monitoring device utilization and switchover of alias devices.

- FastDPAV requires no coordination with other systems.
- Alias devices are available immediately whenever required.

In native mode, BS2000 automatically activates all generated FastDPAV alias devices for I/O.

FastDPAV can be used with or without static PAV.

Fast Dynamic PAV (FastDPAV) in VM2000 operation

In the VM2000 mode, BS2000 activates only those generated FastDPAV alias devices that have been **exclusively** assigned to the VM.

> ¡ VM2000 prevents from a shared assignment (usage mode `SHARED`) for FastDPAV alias devices.

The FastDPAV alias devices can be assigned explicitly (using `/ADD-VM-DEVICES`), or implicitly (using `/ATTACH-DEVICE`) provided the VM and the devices have the required authorization.

When performing I/O, BS2000 considers only the activated FastDPAV alias devices. You can define with BS2000 command `/MODIFY-IO-OPTIONS` whether base or alias devices are to be preferred for starting I/Os in the guest system.

> ¡ The FastDPAV alias devices are not handled by VM2000 as conventional alias devices (PAV/DPAV), thus they are not displayed like those with command `/SHOW-VM-DEVICE-STATUS` either.

Example of the use of (static) PAV in 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)
***
*** Gastsystem-Operating auf der Monitor-VM:
*** /attach-device unit=2300 ──────────────────────────────────────────  (5)
*** NKR0042 'DEVICE     =2300': ATTACH ACCEPTED
*** NKR0042 'DEVICE     =2300': ATTACHED
*** NKR0042 'DEVICE     =2310': ATTACH ACCEPTED ──────────────────────  (6)
*** NKR0042 'DEVICE     =2320': ATTACH ACCEPTED
*** NKR0042 '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: <date>
*** NKR0073 ALIAS DEVICE '2310' LINKED TO BASE DEVICE '2300'
*** NKR0073 ALIAS DEVICE '2320' LINKED TO BASE DEVICE '2300'
*** NKR0073 ALIAS DEVICE '2330' LINKED TO BASE DEVICE '2300'
*** NKR0040 'DEVICE     =2310' ATTACHED
*** NKR0040 'DEVICE     =2320' ATTACHED
*** NKR0040 '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)
/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
***
*** Gastsystem-Operating auf der Monitor-VM:
*** /detach-device unit=2300  ──────────────────────────────────────────── (10)
*** NKR0092 'DEVICE     =2300': /DETACH-DEVICE PENDING ACCEPTED
*** NKR0045 *** 'DEVICE     =2300' DETACHED ***
*** NBR0740 COMMAND COMPLETED 'DETACH-DEVICE';
***            (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
*** NKR0074 ALIAS DEVICE '2310' UNLINKED FROM BASE DEVICE '2300' ──────── (11)
*** NKR0074 ALIAS DEVICE '2320' UNLINKED FROM BASE DEVICE '2300'
*** NKR0074 ALIAS DEVICE '2330' UNLINKED FROM BASE DEVICE '2300'
*** NKR0045 *** 'DEVICE     =2310' DETACHED ***
*** NKR0045 *** 'DEVICE     =2320' DETACHED ***
*** NKR0045 *** '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)
```

---

(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.

(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.

(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 detached implicitly 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 (SU /390)

The Highspeed Network Connect HNC connects an SU /390 with different IP networks. The connection to the networks can be set up via the direct port on the HNC or via the Net Unit. The Net Unit offers a number of networks in the SE server. The HNC is a component of the SE server and is operated and managed via the SE Manager, see the “Operation and Administration” manual [19].

With one HNC, network connections to the monitor system and all other guest systems can be established in VM2000 operation.
The data transfer between the various BS2000 guest systems of an SU /390 takes place in the HNC's internal network, without placing a load on private or customer networks.

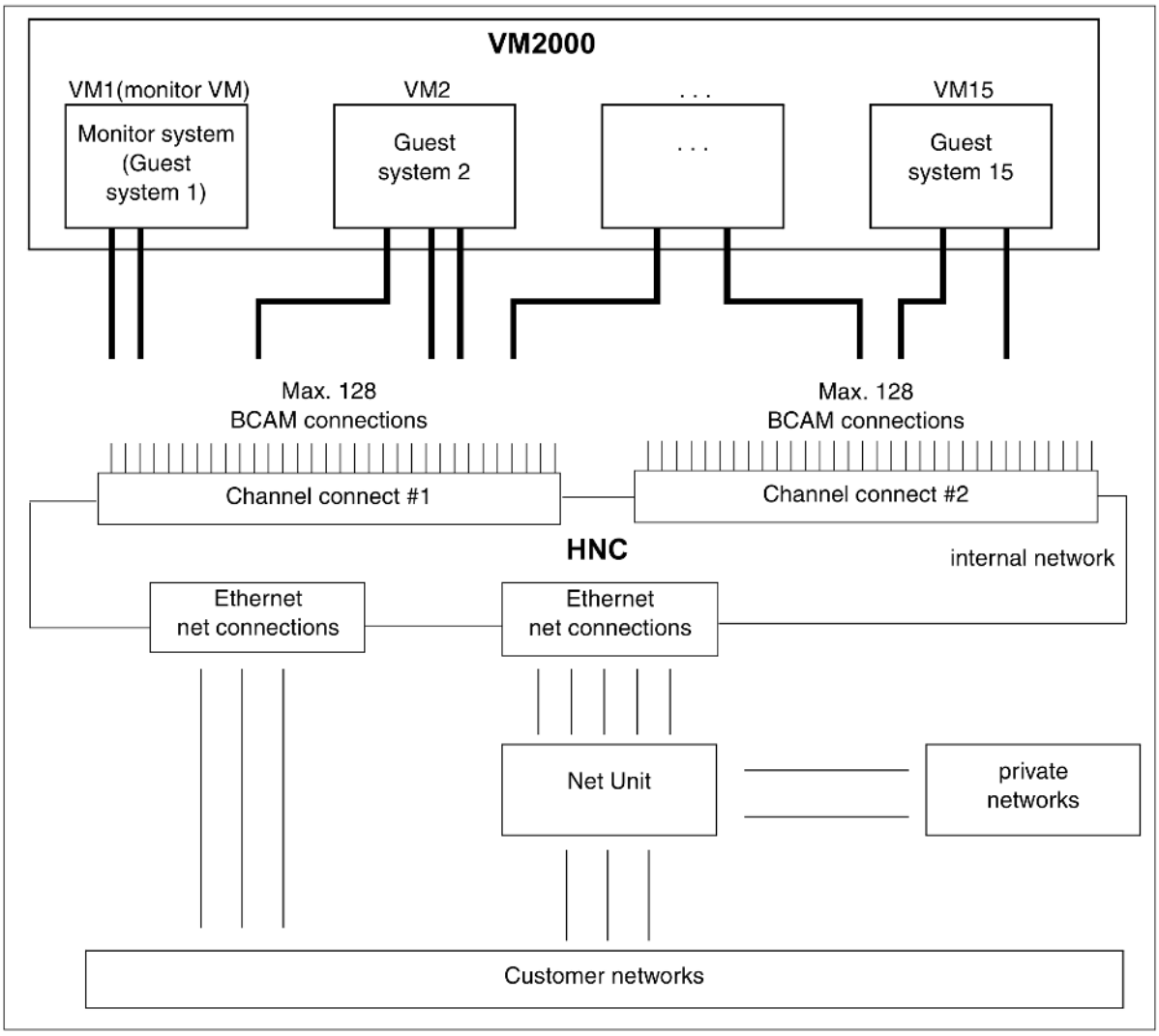


Figure 25: Using the HNC in VM2000

The HNC has 2 channel connects. On each channel connect, 128 BCAM network connections can be configured. Up to 4 HNCs can be connected on an SU /390, for example to set up redundant BCAM network connections. The HNC can be used to set up connections to different networks, e.g. client networks (DANPUx), SE private networks (DANPRxx) or the internal network (LOCLAN). For a successful connection, a BCAM line adapter must be defined for every guest system to the desired network and a corresponding address pair must be set up.

Specimen configuration

In the following example, such a configuration is assumed. The statements needed for installation, generation and assignment are shown.

Three Ethernet networks and the internal network are installed. Three guest systems are running in VM2000 operation:

- The monitor system (`MONIT` in VM1)
- Two production systems (`PROD1` in VM2, `PROD2` in VM3)

An HNC is installed on two channels (channel connect #1 (C0) and #2 (C1)).

The assignment of the device address pairs to VMs, channel connects and networks can be seen from Figure 26.

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).

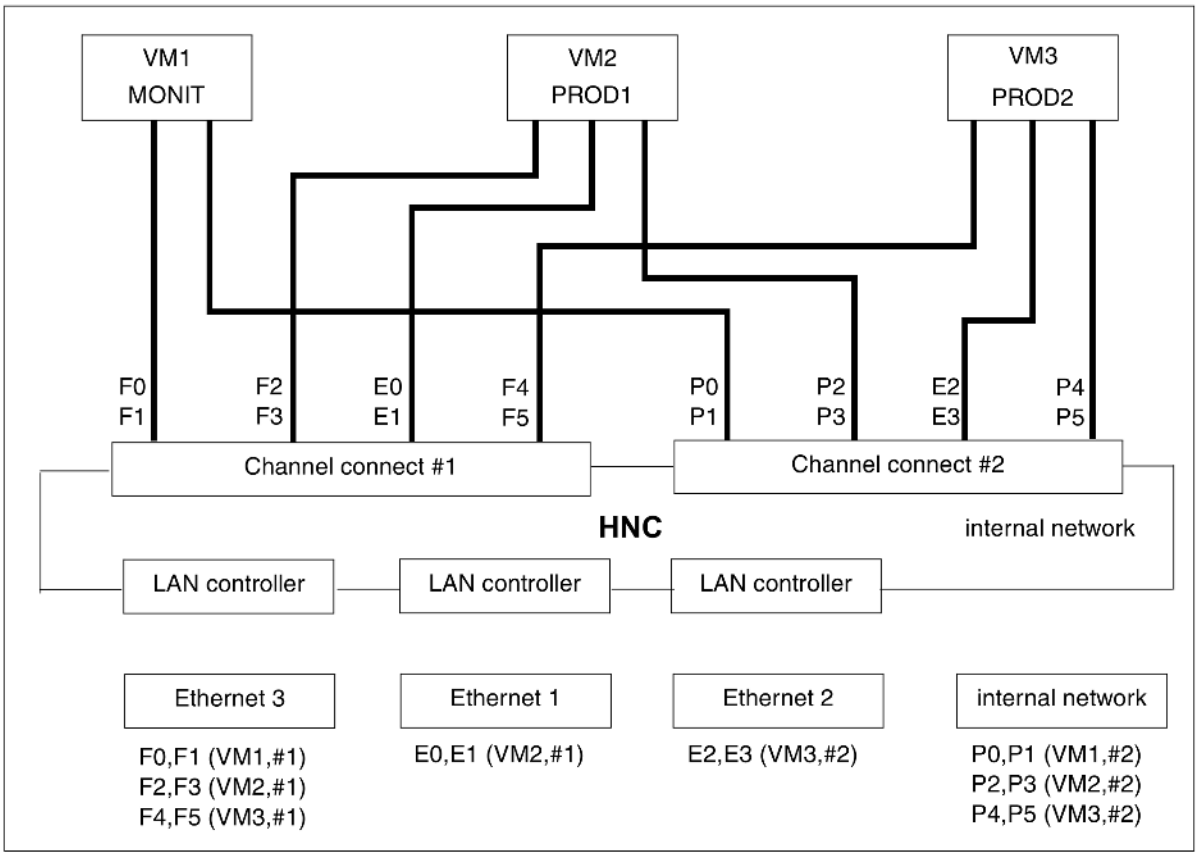


Figure 26: Specimen configuration for HNC

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 Customer 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 connects and for the 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)
* Channel connect #2 of HNC
   CTL C1,,(K1,0)
* ETHERNET3 via channel connect #1 (C0)
   DVC F0,6D,A,00,(C0)
   DVC F1,6D,A,01,(C0)
   DVC F2,6D,A,02,(C0)
   DVC F3,6D,A,03,(C0)
   DVC F4,6D,A,04,(C0)
   DVC F5,6D,A,05,(C0)
* ETHERNET1 via channel connect #1 (C0)
   DVC E0,6D,A,06,(C0)
   DVC E1,6D,A,07,(C0)
* Internal network via channel connect #2 (C1)
   DVC P0,6D,A,00,(C1)
   DVC P1,6D,A,01,(C1)
   DVC P2,6D,A,0C,(C1)
   DVC P3,6D,A,03,(C1)
   DVC P4,6D,A,04,(C1)
   DVC P5,6D,A,05,(C1)
* ETHERNET2 via channel connect #2 (C1)
   DVC E2,6D,A,06,(C1)
   DVC E3,6D,A,07,(C1)
```

The BCAM line adapters to the device address pairs must be defined by using the appropriate BCAM statements (SOF definitions) (see the "BCAM" manual [8]).

Assigning the specimen configuration to VM2000

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 VM 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
```

In the above example, the following statements will be output for the network configuration of the production systems after setting up the VMs:

```
/ADD-VM-DEVICES UNITS=(F2,F3,E0,E1,P2,P3),VM-ID=PROD1
/ADD-VM-DEVICES UNITS=(F4,F5,E2,E3,P4,P5),VM-ID=PROD2
```

6.4.7 Notes on configuration for VM2000 operation

In theory the peripheral devices belonging to a Server Unit 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 Units

- The operation of the guest system on a VM (operating) is normally carried out by means of BS2000 consoles (see "Operating the guest systems using BS2000 consoles").

  Alternatively guest systems can also be operated via $VMCONS.

Notes on configuration for the SU x86

- In each guest system (and the monitor system as well), the configuration states 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 the SU /390

- If all KVP devices for guest systems have been generated as DETACHED (recommended) during hardware generation, and there are also no redundant KVP devices (KVP devices of the redundant MU that are assigned to a VM but are not being used as an IPL console), the configuration states of the redundant KVP devices in every guest system should be set to ATTACHED in the startup parameter service (IOCONF parameter service, MODIFY-IO-UNIT statement).
- If different virtual machines are working with channel devices that are attached to **one** device controller, a very 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.
- In VM2000 operation, disk devices can be assigned to the VMs for shared use. If a disk device is shared by at least two VMs on an SU /390, all of the input/output traffic of the shared disk devices is coordinated by the VM2000 hypervisor. On SU /390, disk devices should therefore not be operated as shareable unnecessarily.

6.4.8 Use of IORM in VM2000 operation

The BS2000 utility routine IORM (I/O 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 operation with the storage system ETERNUS CS
- TCOM: Dynamic Tape Compression
- IOLVM: I/O Limit for Virtual Machines

x86 The IORM functions DPAV and IOLVM have no relevance for SU x86.

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 Unit 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 Unit:

- IOPT
  The IOPT function is enabled on the guest systems on which IOPT is to be active.
- DPAV (SU /390)
  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 "Parallel access volume (PAV, SU /390)".
- DDAL
  This function extends the optimized (local) device selection in native mode to all guest systems of a Server Unit 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.
- TCOM
  The TCOM function is enabled on the guest systems on which TCOM is to be active.

- IOLVM (SU /390)
  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 then 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 when the VM is initialized (see "Initializing a VM") or with `/MODIFY-VM-ATTRIBUTES.`

  > i `/CREATE-` or `/MODIFY-VM-DEFINITION` enables the I/O limit for a VM to be entered in the VM definition. Detailed information can be found in the section "Working with VM definitions".

  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

- Changing the configuration of channel devices using BS2000 (SU /390)
- Changing the configuration of devices via X2000 (SU x86)
- Changing the configuration of devices in the disk storage system

6.4.9.1 Changing the configuration of channel devices using BS2000 (SU /390)

On SU /390, 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 Unit and thus take effect immediately.

x86 No channels are available on SU x86.

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 “Introduction to System Administration” [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 or in a VM Migration. On the other hand, a guest system cannot be started (`/START-VM`) or migrated (`/MIGRATE-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 configuration description 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 "Reconfiguring Channels, controllers and paths (SU /390)").

The information commands of VM2000 and of the device management facility in BS2000 provide information on the current configuration.

6.4.9.2 Changing the configuration of devices via X2000 (SU x86)

The configuration of the devices managed on the SU x86 by X2000 can be changed during ongoing VM2000 operation using the SE Manager, see the “Operation and Administration” manual [19].

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 can be attached again 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 Customer Service, none of the devices involved may be assigned to a VM.

6.4.10 Checking and updating of the VM2000 disk configuration (SU /390)

The VM2000 disk configuration can also change during ongoing VM2000 operation, e.g. by adding new or reconfiguring existing disk devices. Therefore, checking and updating the VM2000 disk configuration may become necessary even after starting VM2000.

In the following cases, VM2000 automatically checks and updates the disk configuration of affected disk devices:

- a dynamic I/O configuration modification (see section "Dynamic I/O configuration modification")
- every exclusive assignment of a disk device to a VM or an assignment of a shareable disk device to the first VM using `/ADD-VM-DEVICES`
- a request for the hardware data of a disk device by a guest system (for SHC-OSD), if the device is yet unknown in VM2000

In some cases, VM2000 is unable to check the disk configuration automatically.

The VM2000 administrator can manually check and update the VM2000 disk configuration with the `/CHECK-VM-DISK-CONFIGURATION` command.

In the following cases, it can be helpful to check and update the VM2000 disk configuration manually with `/CHECK-VM-DISK-CONFIGURATION`:

- When VM2000 is started, there are already disk devices defined in the IOCF (IORSF) but not configured in the disk storage system. These disk devices can be configured in the disk storage system at a later point.
- Free devices in the disk storage system have been reconfigured there and they have probably received different internal device numbers.

In both cases, `/CHECK-VM-DISK-CONFIGURATION` identifies the most recent hardware data and the PAV configuration of these disk devices and then updates the VM2000 disk configuration. These devices do not have to be assigned to a VM.

`/CHECK-VM-DISK-CONFIGURATION` initiates checking the VM2000 disk configuration (`VMS4910` message). The VM2000 hypervisor checks the data of all generated disk devices and updates the data of the VM2000 disk configuration. The completion of the check is reported asynchronously to the VM2000 administrator and the console of the monitor system with the `VMS4912` message.

In the above two cases, transfer of the correct hardware data to SHC-OSD is only ensured with `/CHECK-VM-DISK-CONFIGURATION`.

6.5 Backing up and restoring the VM2000 configuration

By saving the current VM2000 configuration, the VM2000 administrator is able to restore the saved VM2000 configuration after a (re-)start of the whole VM2000 system.

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 Unit (e.g. a backup) on which no BS2000 guest systems run except the monitor system.

- Backing up a VM2000 configuration
- Modifying the backup of a VM2000 configuration
- Restoring a VM2000 configuration
- Backup file for a VM2000 configuration
  - Storage location
  - Structure of the contents
  - VM2000 components not backed up
  - Example of a backup file (SU /390)

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.

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".

Some components of the VM2000 configuration are not backed up, see section "VM2000 components not backed up".

Persistent VMs are not saved because their persistent VM definitions are also available after the Server Unit has been rebooted and can be reactivated with /ACTIVATE-VM-DEFINITION. Therefore, for each persistent VM a warning (VMS4817) is issued in the case of /SAVE-VM-CONFIGURATION.

A VM2000 configuration can only be backed up successfully if the prerequisites described below are fulfilled at command entry time:

- No VM may be in state IN HOLD (WAIT)
- All VMs in state RUNNING must have reached SYSTEM READY
- Memory reconfiguration may not be in progress
- VM Migration may not be in progress

Otherwise the command is 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.

Nevertheless, if the file is to be edited, 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".

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 being impossible 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.

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") must, if necessary, be restored manually.

A VM2000 configuration can only be restored successfully if the requirements described below are fulfilled at command entry time.

If one of the requirements is not fulfilled, the command is rejected or continued. The consequences of continuing the command are described for the respective requirements below.

Requirements:

- 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.
- Requirements for the real CPUs:
  - The number of operable normal CPUs (states ATT or DET) 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 *STDPOOL but also another VM2000 CPU pool, the command is rejected with the message VMS4812.
    2. If in addition to the CPU pool *STDPOOL 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 *STDPOOL but also another VM2000 CPU pool, the command is rejected with the message VMS4812.
    2. If in addition to the CPU pool *STDPOOL 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
- The VM name of a VM from the backup may not be the same as the VM name of an existing persistent VM definition

- Requirements for the current monitor VM:
  - Memory reconfiguration may not be in progress
  - The VM name of the current monitor VM is not assigned to any other VM than the monitor VM in the backed-up configuration

    Otherwise, the warning message VMS4816 is output and processing is continued. If in the further processing a VM cannot be configured because of the VM name, message VMS4803 is output for this VM.
  - On SU /390, 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 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 SU x86, 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 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 requirement is not checked. If in the 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 to an assignment set in the backed-up VM2000 configuration must be available to VM2000 again in the current system

  This requirement is not checked. If in the 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.

- Storage location
- Structure of the contents
- VM2000 components not backed up
- Example of a backup file (SU /390)

6.5.4.1 Storage location

The VM2000 administrator can define the name of the backup file explicitly. In doing so, a foreign user ID may be specified for storing the backup file only if the VM2000 administrator has the TSOS privilege.

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.
- Command entry from a KVP console or a logical console
  - Without operator LOGON
    The file name is completed with the TSOS user ID.
  - With operator LOGON
    The file name is completed with the operator user ID.
- 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 "Setting up the VM2000 monitor").

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 <*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>

XML tag	Meaning	Possible values
<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>
<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>	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 guest system on the VM IPL device for the start of the guest system on the VM Output path of KVP console when starting the guest system on the VM Input path of KVP console when starting the guest system on the VM End of first BS2000 VM Further VMs (structure as before) End of BS2000 VMs	 <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> / *MAXIMUM / *EXTRA-AND-NORMAL <integer 1..9999> <integer 1..10000> <x-text 1..8> *STDPOOL / <name 1..8> <integer 1..100> *MAXIMUM / *MINIMUM / *STANDARD *AT-DEDICATED-CPUS / *NO *NO / *YES *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>

XML tag	Meaning	Possible values

<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>
	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
		
		<alphanum-name 2..2> /
		<x-text 4..4>
		*NO / *YES
		*NO / *YES
		*STDSET / <name 1..8>
		EXCL / SHARED / VC
		
		<integer 1..99>
		*NO / *YES

</vm2000_config>	End of VM2000 configuration data	

6.5.4.3 VM2000 components not backed up

The following components of the VM2000 configuration are **not** backed up:

- Passwords of $VMCONS
- MONJVs
- VM definitions
- Persistent VMs
- 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 states 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 (SU /390)

For the sake of clarity some outputs in the example below are abbreviated (". . .").

```
<vm2000_config>
  <global_data>
    <vm2000_version>V12.0B00</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>15872</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>512</min_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>
  <vm>
    <vm_index>2</vm_index>
    <vm_name>G4IVU</vm_name>
    <mem_size>512</mem_size>
    <min_mem_size>512</min_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_privileg>*STANDARD</admin_privileg>
    <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 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 the 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 or after a migration of the VM.

> ¡ On SU /390, the password for administering the monitor VM can be defined via the configuration file for the monitor VM (see "Configuration via the VM2000 configuration file (SU /390)") and on SU x86 via the configuration of VM2000 in the SE manager (see "Configuration using the SE Manager (SU x86)").
>
> `/CREATE-` or `/MODIFY-VM-DEFINITION` enables the password for a VM to be entered in the VM definition. Detailed information can be found in the section "Working with VM definitions".

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 "Setting up the VM2000 monitor"). 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`. 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.

6.7 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 "Assignment sets, implicit device assignment and release")
  - `/SHUTDOWN` in the monitor system (`/DELETE-VM` for all VMs)
  - A restart in the monitor system (implicitly assigned devices are released before `SYSTEM READY`, see "Assignment sets, implicit device assignment and release")
  - VM Migration (on the source SU)

  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`
- 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`
- VM Migration (on the source SU)

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 “Introduction to System Administration” [2]). Therefore, the accounting system in the monitor system should be always active.

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

6.7.1 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 section (format: yymmdd)
2	06	6	Z	Time of release of devices (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 section (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 section: 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

6.7.2 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 "Reconfiguring main memory")

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.

6.8 Hardware error logging in 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 well as in VM2000 mode 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. In VM2000 mode, 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 in VM2000 operating on a Server Unit, 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.

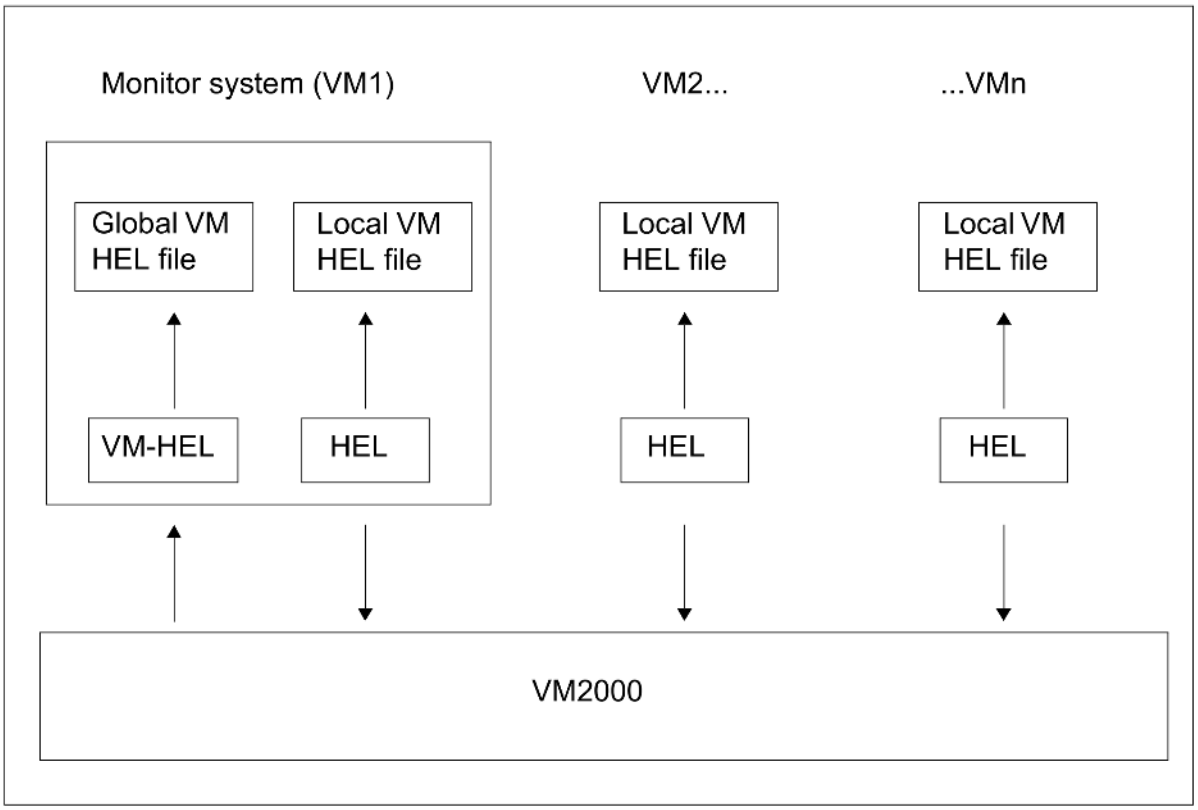


Figure 27: Concept of cross-VM hardware error logging under VM2000

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 in the VM2000 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 HWMAINT (HARDWARE-MAINTENANCE). As for the command CHANGE-(VM)HEL-FILE, also the system privilege TSOS is sufficient. The VMHEL commands have the same syntax as the HEL commands. The HEL commands are described in detail in the “Commands” manual [13].

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`.

6.9 Monitoring in VM2000

The following monitoring options are available in VM2000:

- Monitoring in 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 restriction governs both monitoring options:

- IDLE times cannot be monitored precisely in a VM2000 guest system

6.9.1 Measurement with /SHOW-VM-STATUS

The command /SHOW-VM-STATUS gives the VM2000 administrator an overview of the current measured values for active objects in VM2000. It 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.

x86 The operand values INFORMATION=*COUNTER/*SCHEDULE are not available on SU x86.

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 (from 5 seconds up 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.

6.9.2 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 [9]. The peculiarities and interpretations of the measured values in VM2000 operation 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 SU /390:

- 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
  - with the VM2000 commands `/DELETE-VM`, `/MIGRATE-VM` (on the source SU), `/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). In this case, SM2 issues a warning message and can only record the `DEVICE QUEUE` and `REMAINING SERVICE TIME`.

7 VM2000 commands

This chapter contains a description of all VM2000 commands. The command syntax is explained in the "Commands" manual [13].

The section section "VM2000 Operating strategy" 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 "Privileged functions".

Notational conventions

The command descriptions contain a great deal of information which only applies for SU /390 or SU x86. These are labeled as follows:

390 This comment applies only for SU /390

x86 This comment applies only for SU x86

- Overview of commands for VM2000 operation
- Command return codes
- Guaranteed messages
- SHOW output in S variables
- Description of the commands

7.1 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 or reduced for the VM administrator or the VM2000 administrator respectively (`ADMIN-PRIVILEGES`).

7.1.1 Table of VM2000 commands

The following table lists the VM2000 commands in alphabetical order.

Command name	VM2000 administrator	VM administrator	Privileged operands (VM-ID=)
ACTIVATE-VM-DEFINITION	x		
ADD-VM-DEVICES	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]	
CHECK-VM-DISK-CONFIGURATION	x		
CHECK-VM-MIGRATION	x		
CREATE-VM	x		
CREATE-VM-ASSIGNMENT-SET	x		
CREATE-VM-CPU-POOL	x		
CREATE-VM-DEFINITION	x		
CREATE-VM-GROUP	x		
DELETE-VM	x	x	*ALL
DELETE-VM-ASSIGNMENT-SET	x		
DELETE-VM-CPU-POOL	x		
DELETE-VM-DEFINITION	x		
DELETE-VM-GROUP	x		
DETACH-VM-RESOURCES	x		
END-VM-DIALOG	x	x	
EXTEND-VM-MEMORY	x		
HOLD-VM	x	x	*ALL

Command name	VM2000 administrator	VM administrator	Privileged operands (VM-ID=)
MIGRATE-VM	x		
MIGRATE-VM-DEFINITION	x		
MODIFY-VM-ATTRIBUTES	x	x [1]	
MODIFY-VM-DEFINITION	x		
MODIFY-VM-DEVICE-ATTRIBUTES	x		
MODIFY-VM-DEVICE-USAGE	x	x [1]	
MODIFY-VM-GROUP-ATTRIBUTES	x		
MODIFY-VMCONS-PARAMETERS	x		
MOVE-VM	x		
RECOVER-VM-DEFINITION	x		
REDUCE-VM-MEMORY	x		
REMOVE-VM-DEVICES	x	x	
REMOVE-VM-FROM-VM-GROUP	x		
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-DEFINITION	x		
SHOW-VM-DEVICE-STATUS	x	x [1]	

Command name	VM2000 administrator	VM administrator	Privileged operands (VM-ID=)
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 "Extending and restricting the range of commands/functions")

2 Command available only in ADMIN dialog via $VMCONS

7.1.2 Grouping of VM2000 commands according to function

Operating a VM

Command	Function	Authorization	Privileged operands (VM-ID=)
CREATE-VM	Initializes a VM (the VM is assigned attributes and resources)	VM2000 Administrator	
MODIFY-VM-ATTRIBUTES	Modifies VM attributes	VM2000 Administrator [1]	
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	
MIGRATE-VM	Migrates a VM	VM2000 Administrator	
CHECK-VM-MIGRATION	Checks if a VM can be migrated	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 "Extending and restricting the range of commands/functions")

Working with VM definitions

Command	Function	Authorization
CREATE-VM-DEFINITION	Creates a VM definition	VM2000 administrator
ACTIVATE-VM-DEFINITION	Activates a VM definition	VM2000 administrator
MIGRATE-VM-DEFINITION	Migrates a VM definition	VM2000 administrator
MODIFY-VM-DEFINITION	Modifies a VM definition	VM2000 administrator
SHOW-VM-DEFINITION	Shows attributes of VM definitions	VM2000 administrator
RECOVER-VM-DEFINITION	Recovers a VM definition	VM2000 administrator
DELETE-VM-DEFINITION	Deletes a VM definition	VM2000 administrator

Operating VM groups (SU /390)

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	Creates an assignment set	VM2000 administrator
DELETE-VM-ASSIGNMENT-SET	Deletes an assignment set	VM2000 administrator
SHOW-VM-ASSIGNMENT-SET	Shows attributes of the assignment sets	VM2000 administrator
ADD-VM-DEVICES	Allocates available devices to a VM	VM2000 administrator [1]

Command	Function	Authorization
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	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 "Extending and restricting the range of commands/functions")

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	Removes CPUs from a CPU pool and assigns them to another one	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
CHECK-VM-DISK-CONFIGURATION	Checks the VM2000 disk configuration	VM2000 administrator
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	Detach real CPUs	VM2000 administrator

[1] Command available to VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM`
(see "Extending and restricting the range of commands/functions")

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

7.2 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 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	Main code	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 [13].

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 Units and are not supported on them, see also paragraph “Notational conventions” in section "VM2000 commands". If they are entered despite this, they are rejected by VM2000 with a message and a return code.

Entry	Message	Return code
Command	VMS3011	0, 64, VMS9004
Operand	VMS3012 VMS3030	0, 64, VMS9004
Operand value	VMS3032	0, 64, VMS9004

7.3 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 in the description of the relevant commands in paragraph “Guaranteed messages“.

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 output for the information of the list elements processed.

For information on working with S variables, see the “Commands“ manual [13].

7.4 SHOW output in S variables

VM2000 supports a structured output in S variables for all its information commands (with the exception of `/SHOW-VM-STATUS)` if they are executed in a user task. The S variables can be analyzed with the software product SDF-P.

A structured output means that the output information is supplied in a list variable (of type structure) that is declared by the user. Every element of the structure may be either a simple S variable, or a complex one again. At the hierarchical lowest level, there are simple S variables of type S (string), I (integer) or B (boolean) for the elementary information.

The layout of the superordinate S variable of type structure (listing all simple S variables for the elementary information) is a component of the respective command description. The simple S variables are sorted in a table according to application groups and listed there under specification of their name, meaning, content and type.

Names of the S variables

The names (e.g. `VM-INDEX`) are preset for the respective information commands. They are appended to the S variable name declared by the user. The names are oriented, as far as possible, toward the corresponding operand names or output designations. If the information at this level continues to be hierarchically subdivided, the names of the lower levels are appended, each separated by a period.

The user can access either the desired elementary information by specifying the name of the corresponding simple S variable, or the complex information by specifying the superordinate S variable.

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.

Contents of the simple S variables

As far as possible, the contents of the simple S variables correspond to the associated output values for SYSOUT. The type of the S variable determines the character set from which the contents of the S variable originate.

Types of the simple S variables

The simple S variables are of type: S (string), I (integer) or B (boolean). The type is presented in column T of the table.

For information on working with S variables, see the “Commands“ manual [13].

For an example of the evaluation of S variables in VM2000, turn to "Automating VM2000 and VM administration".

7.5 Description of the commands

The commands are described alphabetically under the following headings:

- Command name and function
- Description of the command function
- Representation of the command format
- Description of the command operands
- Description of the command return codes
- List of guaranteed messages
- For information commands: output in S variables
- For information commands: output formats
- Example if required

Command name	Function
ACTIVATE-VM-DEFINITION	Activate VM definition
ADD-VM-DEVICES	Allocate devices to a VM
ADD-VM-TO-VM-GROUP	Add a VM to a VM group 390
ASSIGN-VM-GROUP-TO-CPU-POOL	Assign a VM group to a CPU pool 390
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
CHECK-VM-DISK-CONFIGURATION	Checks the VM2000 disk configuration 390
CHECK-VM-MIGRATION	Checks if a VM can be migrated
CREATE-VM	Initialize a VM
CREATE-VM-ASSIGNMENT-SET	Create assignment set
CREATE-VM-CPU-POOL	Create a CPU pool
CREATE-VM-DEFINITION	Create a VM definition
CREATE-VM-GROUP	Create a VM group 390
DELETE-VM	Terminate a VM
DELETE-VM-ASSIGNMENT-SET	Delete an assignment set

Command name	Function
DELETE-VM-CPU-POOL	Delete a CPU pool
DELETE-VM-DEFINITION	Delete a VM definition
DELETE-VM-GROUP	Delete a VM group 390
DETACH-VM-RESOURCES	Detach real CPUs
END-VM-DIALOG	Terminate administration or operation
EXTEND-VM-MEMORY	Increase main memory for a VM
HOLD-VM	Suspend VM 390
MIGRATE-VM	Migrate a VM
MIGRATE-VM-DEFINITION	Migrate a VM definition
MODIFY-VM-ATTRIBUTES	Modify attributes of a VM
MODIFY-VM-DEFINITION	Modify a VM definition
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 390
MODIFY-VMCONS-PARAMETERS	Modify parameters of $VMCONS
MOVE-VM	Move a VM in main memory 390
RECOVER-VM-DEFINITION	Recover a VM definition
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 390
RESET-VM	Aborts the guest system on a VM
RESTORE-VM-CONFIGURATION	Restores the saved VM2000 configuration
RESUME-VM	Resume VM operation 390
SAVE-VM-CONFIGURATION	Saves the current VM2000 configuration
SEND-VM-MESSAGE	Pass messages to VM administrators
SHOW-VM-ASSIGNMENT-SET	Output assignment sets

Command name	Function
SHOW-VM-ATTRIBUTES	Output attributes of a VM
SHOW-VM-CPU-POOL	Output attributes of a CPU pool
SHOW-VM-DEFINITION	Output a VM definition
SHOW-VM-DEVICE-STATUS	Output detailed device information
SHOW-VM-GROUP	Output attributes of a VM group 390
SHOW-VM-RESOURCES	Output status of VM2000 resources
SHOW-VM-STATUS	Output VM2000 monitored data
SHUTDOWN-VM	Execute a shutdown in the guest system and for VM2000
START-VM	Start a guest system on a VM
SWITCH-VM-CPU	Remove CPUs from a CPU pool, add them to another
SWITCH-VM-DEVICES	Deallocate devices from one VM, allocate them to another

7.5.1 ACTIVATE-VM-DEFINITION (Activate a VM definition)

/ACTIVATE-VM-DEFINITION activates a VM definition, i.e. a VM is initialized with the attributes entered in the VM definition. The devices entered in the VM definition are assigned to the VM. If required, the guest system on the VM is started in accordance with the settings for automatic startup in the VM definition. The VM definition may not belong to a VM which has already been initialized.

The VM can be monitored with a monitor job variable.

This command is reserved for the VM2000 administrator.

ACTIVATE-**VM-DEF**INITION
VM-NAME = <name 1..8> **,MONJV**= ***NONE** / <filename 1..54 without-gen-vers> ,**JV-PASS**WORD = ***NONE** / <c-string 1..4> / <x-string 1..8> / <integer -2147483647..2147483648>

VM-NAME = <name 1..8>
Name of the VM definition.
Identical to the name of the virtual machine which is to be initialized.

MONJV =
Defines a MONJV for monitoring the VM (see "Monitoring a VM with job variables (MONJV)").

MONJV = *NONE
No MONJV is defined.

MONJV = <filename 1..54 without-gen-vers>
The specified MONJV is supplied with data by VM2000; therefore, the sender of the command must 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 the specified file name does not refer to an existing job variable, 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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS2306	VM definition could not be updated (last IPL parameter in a definition with AUTO-IPL=*YES)
	64	VMS2101	VM2000 shutdown in progress
	64	VMS2300	VM definition does not exist
	64	VMS2303	VM definition cannot be accessed
	64	VMS2314	Conflict in the device usage mode between the pubset and the PUBRES device
	64	VMS2316	VM index greater than the current maximum number of VMs
	64	VMS2318	VM index is already used by an initialized VM
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	64	VMS3035	IO-PRIORITY cannot be set (total of virtual CPUs with privilege too high)
	64	VMS4004	Device already assigned to a VM
	64	VMS4006	Device not available (not defined)
	64	VMS4010	Maximum permitted number of VMs already reached
	64	VMS4011	Requested memory size or location not available
	64	VMS4014	SVP error
	64	VMS4015	VM already initialized
	64	VMS4016	Device cannot be used as 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	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	VMS4200	Assignment set not yet created
	64	VMS4250	Device is not a PUBRES device
	64	VMS4251	Device list of the pubset is inconsistent
	64	VMS4252	Device list for PUBRES cannot be determined
	64	VMS4340	MONJV could not be accessed
	64	VMS4400	VM group not yet created
	64	VMS4513	Hypervisor memory bottleneck
	64	VMS4600	CPU pool not yet created
	64	VMS4612	No attached CPU in the CPU pool

Guaranteed messages

`VMS4005`

Only on SU /390: `VMS3050, VMS4425, VMS4615`

Only on SU x86: `VMS6020`

Example

See section "Examples of working with VM definitions".

7.5.2 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.

The assigned devices are noted in the VM definition.

No devices can be assigned explicitly to a VM while it is being migrated (VM Migration).

> The devices must be then 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.

390 Alias devices (PAV/DPAV) and CRD devices cannot be assigned explicitly.

This command is reserved for the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM`.

ADD-VM-DEVICES

UNITS = ***PUB**SET**-DEV**ICES(...) /

list-poss(28): ***DEVICE-RANGE**(...) / <alphanum-name 1..4 with-wild(20)>

***PUB**SET**-DEV**ICES(...)

| **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| ***BY-PUBRES-DEV**ICE(...)

| | **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-ID**ENTIFICATION = <integer 1..99> / <name 1..8>

,**TYPE** = **<u>*NONE</u>** / ***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" and "Pubset reconfiguration".

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.

390 If UNITS are specified with *DEVICE-RANGE or with wildcards, alias devices (PAV/DPAV) and CRD devices are ignored.

x86 Due to the performance reasons, the assignment of a large amount of devices with /ADD-VM-DEVICES (UNITS specification with *DEVICE-RANGE or wildcards) should be avoided on SU x86. The device assignment should be accomplished preferably before the activation of a persistent VM definition or via the SE Manager.

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 "Shared disk devices"). Shared devices are listed in the information commands after the keyword SHARED.

TYPE = *VC

The device is assigned exclusively as a virtual console (SU /390). It must be appropriately generated (see "Device peripherals on SU /390"). 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 codes

(SC2)	SC1	Main code	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	VMS2306	VM definition could not be updated (SU /390)
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/DPAV)
	64	VMS4195	Device is a CRD device
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command not or only partially processed
	128	VMS3042	VM being migrated

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, VMS4195

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 thus identified by VM2000.

(3) The VM2000 administrator assigns the KVP device pair to VM2.

(4) A disk device 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 ("Example of peripheral device management" and "Example of pubset management").

7.5.3 ADD-VM-TO-VM-GROUP (Add a VM to a VM group)

x86 No VM groups are available on SU x86. 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.

The assignment of the VM to a VM group is noted in its VM definition.

A VM cannot be assigned to a VM group while it is being migrated (VM Migration).

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.

This command is reserved for the VM2000 administrator.

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 "Operating VM groups (SU /390)").

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
2	0	VMS2306	VM definition could not be updated (SU /390)
	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
	128	VMS3042	VM being migrated

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".

7.5.4 ASSIGN-VM-GROUP-TO-CPU-POOL (Assign a VM group to a CPU pool)

x86 No VM groups are available on SU x86. 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. The VM group is assigned a (new) CPU quota. Together with the VM group, the VMs which are assigned to the VM group are also assigned to the (different) CPU pool.

A VM group cannot be assigned to another CPU pool while a VM in the group is being migrated (VM Migration).

This command is reserved for the VM2000 administrator.

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 CPU pool to which the VM group is to be assigned.

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 "Planning distribution of the CPU capacity to the VMs").

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 "Number of attached real and virtual CPUs in the CPU pool". If a VM in the VM group does not fulfill this condition, the command is rejected.

ATTACHED-VM-CPUS = *ADJUST-NUMBER
The command is executed unless a VM in the VM group is in state RUNNING or IN HOLD (WAIT) 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 marked accordingly by VM2000 (state 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 state BLOCK 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 codes

(SC2)	SC1	Main code	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
	128	VMS3042	VM of the VM group is being migrated

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'
```

7.5.5 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.

The assignment of the VM to a CPU pool is noted in its VM definition.

A VM cannot be assigned to another CPU pool while it is being migrated (VM Migration).

x86 On SU x86, the command is not executed if the CPU pool to which the VM is to be assigned contains no attached CPU.

This command is reserved for the VM2000 administrator.

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).

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 "Number of attached real and virtual CPUs in the CPU pool". Otherwise the command is rejected.

ATTACHED-VM-CPUS = *ADJUST-NUMBER
The command is executed unless the VM is in state RUNNING or IN HOLD (WAIT) 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 marked accordingly by VM2000 (state 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 is less than its multiprocessor level:

  VM2000 cancels state BLOCK 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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
2	0	VMS2306	VM definition could not be updated (SU /390)
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

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".

7.5.6 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 reserved for the VM2000 administrator.

ATTACH-**VM-RES**OURCES

UNIT = ***CPU**(...) / ***EXTRA-CPU**(...)

***CPU**(...)

| **CPU-ID**ENTIFICATION = ***ANY**(...) / list-poss(63): <x-text 1..2>

| ***ANY**(...)

| | **CPU-POOL-NAME** = ***UNIQUE** / ***STDPOOL** / <name 1..8>

***EXTRA-CPU**(...)

| **CPU-ID**ENTIFICATION = ***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 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 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 "Increasing capacity with extra CPUs".

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
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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	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	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`

Example

```
/show-vm-resources information=*cpu ──────────────────────────────────────  (1)
VM-ID        CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
 1 M4IVR     *STDPOOL    20.00    1.00/ 12.04  80.00/ 80.00  2(+1)
 2 G4IVQ     *STDPOOL           26.00/ 15.66 100.00/100.00  4(+1)
 3 G4IVP     *STDPOOL    50.00    4.00/ 24.09 100.00/ 50.00  2(+1)
 4 G4IVO     *STDPOOL    50.00    1.00/  6.02 100.00/ 50.00  1(+1)
 5 G4IVJ     *STDPOOL           70.00/ 42.16 100.00/100.00  2(+1)
VM-ID        SCHED ACT-I  VIRT CPUS
 1 M4IVR      TS    NO    00(RUN),01(RUN),02(OFF)
 2 G4IVQ      TS    NO    00(RUN),01(RUN),02(BLOCK),03(BLOCK),
                          04(OFF)
 3 G4IVP      TS    NO    00(RUN),01(RUN),02(OFF)
 4 G4IVO      TS    NO    00(RUN),01(SLEEP)
 5 G4IVJ      -     NO    00(IDLE),01(IDLE),02(OFF)
CPU-ATTR     CPU-POOL     REAL CPUS
NORMAL       *STDPOOL     00(ATT),01(DET)
EXTRA        *STDPOOL     02(DET),03(DET) ─────────────────────────────────  (2)
SPARE                     04(DET)
/attach-vm-resources unit=*cpu(cpu-id=01) ────────────────────────────────  (3)
VMS4044 CPU 01 ATTACHED
/attach-vm-resources unit=*extra-cpu(cpu-id=*all) ────────────────────────  (4)
VMS4075 EXTRA-CPU 02 ATTACHED
VMS4075 EXTRA-CPU 03 ATTACHED
/show-vm-resources information=*cpu ──────────────────────────────────────  (5)
VM-ID        CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
 1 M4IVR     *STDPOOL    20.00    1.00/ 12.04  80.00/ 80.00  2(+1)
 2 G4IVQ     *STDPOOL           26.00/ 15.66 100.00/100.00  4(+1)
 3 G4IVP     *STDPOOL    50.00    4.00/ 24.09 100.00/ 50.00  2(+1)
 4 G4IVO     *STDPOOL    50.00    1.00/  6.02 100.00/ 50.00  1(+1)
 5 G4IVJ     *STDPOOL           70.00/ 42.16 100.00/100.00  2(+1)
VM-ID        SCHED ACT-I  VIRT CPUS
 1 M4IVR      TS    NO    00(RUN),01(RUN),02(OFF)
 2 G4IVQ      TS    NO    00(RUN),01(RUN),02(STOP),03(STOP),
                          04(OFF)
 3 G4IVP      TS    NO    00(RUN),01(RUN),02(OFF)
 4 G4IVO      TS    NO    00(RUN),01(SLEEP)
 5 G4IVJ      -     NO    00(IDLE),01(IDLE),02(OFF)
CPU-ATTR     CPU-POOL     REAL CPUS
NORMAL       *STDPOOL     00(ATT),01(ATT)
EXTRA        *STDPOOL     02(ATT),03(ATT) ─────────────────────────────────  (6)
SPARE                     04(DET)
```

---

(1) The VM2000 administrator requests information on the CPU state.

(2) Real normal CPU `01` and the extra CPUs are detached.

(3) Normal CPU `01` 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".

7.5.7 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 carried out concurrently for each VM from different privileged user tasks. In addition, one ADMIN dialog can be carried out via $VMCONS.
- for operating the guest system via $VMCONS (VC dialog).
  Up to eight VC dialogs can be carried out concurrently on each VM via $VMCONS.

  Guest system operation with BS2000 consoles is described on "Operating the guest systems using BS2000 consoles".

/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-DIA**LOG
VM-IDENTIFICATION = <integer 1..99> / <name 1..8> ,**PASS**WORD = ***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 which was defined when the VM was initialized (see "Initializing a VM") or with /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:

390 An ADMIN dialog and a VC dialog are to be started.
x86 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 codes

(SC2)	SC1	Main code	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: %  VMS1033 PARTNER 'OMS120   D016ZE04' HAS STARTED A VC DIALOG WITH
                  VIRTUAL MACHINE (2,SYSTEM2) ——————————————————————————  (2)
VM01: %  VMS1013 PARTNER 'OMS120   D016ZE04' HAS STARTED AN ADMINISTRATION
                 DIALOG WITH VIRTUAL MACHINE (2,SYSTEM2).
                  ADMINISTRATION-ID USED: @002
VM02:VMS3000 VM2000 COMMAND /BEGIN-VM-DIALOG FOR
     VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
VM02:NBR0740 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.

7.5.8 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 permitted:

- /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.

The procedure file must be cataloged as a SAM or ISAM file. It must have been generated in standard file format, i. e. with a variable record format and with BLKCTRL=DATA and a maximum record length of 2032 bytes. This format is generated by default when the editor EDT is used.

Continuation lines

If a command is longer than one line, 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
FILE**-NAME** = <filename 1..54 without-gen-vers> ,**VM-ID**ENTIFICATION = ***CURR**ENT / <integer 1..99> / <name 1..8> ,**LIST** = ***Y**ES / ***NO**

FILE-NAME = <filename 1..54 without-gen-vers>
File name of the procedure file.

VM-IDENTIFICATION =
Identification of the virtual machine. The VM must already have been initialized.

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 case of all commands in the procedure file where the VM-IDENTIFICATION operand is **mandatory**, the VM-IDENTIFICATION 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 codes

(SC2)	SC1	Main code	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	Command not allowed 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-IDENTIFICATION=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 identification `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.

7.5.9 CHECK-VM-DISK-CONFIGURATION (Check VM2000 disk configuration)

x86 On SU x86, the command is rejected with the message VMS3011 and the general return code VMS9004.

/CHECK-VM-DISK-CONFIGURATION initiates checking the VM2000 disk configuration.

The VM2000 hypervisor checks the data of all generated disk devices and updates the data of the VM2000 disk configuration.

The completion of the check is reported asynchronously to the VM2000 administrator and the console of the monitor system with the VMS4912 message.

This command is reserved for the VM2000 administrator.

CHECK-VM-DISK-CONFIGURATION

This command does not have operands.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	Checking already initiated, no action
	64	VMS2101	VM2000 shutdown in progress

Guaranteed messages

VMS4910, VMS4911, VMS4912

Example (via the console of the monitor system)

```
/check-vm-disk-configuration ──────────────────────────────────────  (1)
%VM2M-000.134022  %  VMS4910 CHECK OF DISK CONFIGURATION STARTED
+VM2M-000.134022  %  VMS3000 VM2000 COMMAND /CHECK-VM-DISK-CONFIGURATION FOR
VIRTUAL MACHINE ( ,*VM2000) PROCESSED SUCCESSFULLY
! UCO-000.134022  %  NBR0740 COMMAND COMPLETED 'CHECK-VM-DISK-CONFIGURATION';
(RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
/check-vm-disk-configuration ──────────────────────────────────────  (2)
+VM2M-000.134108  %  VMS4911 CHECK OF DISK CONFIGURATION ALREADY STARTED
+VM2M-000.134108  %  VMS3000 VM2000 COMMAND /CHECK-VM-DISK-CONFIGURATION FOR
VIRTUAL MACHINE ( ,*VM2000) PROCESSED SUCCESSFULLY
! UCO-000.134108  %  NBR0740 COMMAND COMPLETED 'CHECK-VM-DISK-CONFIGURATION';
(RESULT: SC2=001, SC1=000, MC=CMD0001); DATE: <date>
%VM2G-000.134114 % VMS4912 CHECK OF DISK CONFIGURATION COMPLETED   ────  (3)

----------------------------------------------------------------------------------------------------
```

(1) Checking of the disk configuration is being initiated.

(2) Checking is not complete.

(3) Asynchronous message after checking.

7.5.10 CHECK-VM-MIGRATION (Checking prerequisites for VM Migration)

/CHECK-VM-MIGRATION checks, if the specified VM can be migrated from the local Server Unit (source SU) to the required Server Unit (target SU).

This command is reserved for the VM2000 administrator.

CHECK-VM-MIGRATION
VM-IDENTIFICATION = <integer 1..99> / <name 1..8> **,SERVER-UNIT-NAME** = ***UNIQUE** / <composed-name 1..64>

VM-IDENTIFICATION =
Identification of the virtual machine.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
Identifies the virtual machine by means of a VM index or VM name.

SERVER-UNIT-NAME =
Determines the Server Unit to which the VM is to be migrated (target SU). The target SU must belong to the same active SU cluster as the local SU (source SU).

SERVER-UNIT-NAME = *UNIQUE
If an SU cluster consists of only two Server Units, the remote SU is unique and chosen automatically as the target SU. Otherwise, the name of the target SU has to be specified.

SERVER-UNIT-NAME = <composed-name 1..64>
Name of the target SU.

> In case of a target SU /390, this is the BS2000 host name of the monitor system.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS2461	VM Migration not possible
	64	VMS2341	No unique remote SU found
	64	VMS2400	Target SU not detected
	64	VMS4000	VM not yet created

Guaranteed messages

VMS2331, VMS2401 through VMS2406, VMS2409, VMS2410, VMS2460, VMS2470, VMS2471, VMS2472

Examples (SU /390)

```
/check-vm-migration vm-id=g4ivo,server-unit-name=d021ze01
%  VMS2404 LIVE MIGRATION OF VM (4,G4IVO) REJECTED BY TARGET SU 'D021ZE01'.
REASON: VMS4011 ──────────────────────────────────────────────────────  (1)
%  VMS2404 LIVE MIGRATION OF VM (4,G4IVO) REJECTED BY TARGET SU 'D021ZE01'.
REASON: VMS4034,9920 ─────────────────────────────────────────────────  (2)
%  VMS2404 LIVE MIGRATION OF VM (4,G4IVO) REJECTED BY TARGET SU 'D021ZE01'.
REASON: VMS2437,9921 ─────────────────────────────────────────────────  (3)
%  VMS2461 LIVE MIGRATION OF VM (4,G4IVO) TO TARGET SU 'D021ZE01' NOT
POSSIBLE ──────────────────────────────────────────────────────────────  (4)
```

The checks on the target SU have delivered the following results:

(1) Not enough main memory for the VM to be migrated.

(2) Device 9920 has already been assigned for shared usage.

(3) Device 9921 has already been assigned exclusively to another VM.

(4) The VM Migration is not possible.

```
/check-vm-migration vm-id=g4ivo,server-unit-name=d021ze01
%  VMS2471 DEVICE '9940' WILL BE DETACHED BY LIVE MIGRATION
%  VMS2471 DEVICE '9941' WILL BE DETACHED BY LIVE MIGRATION
%  VMS2471 DEVICE '9942' WILL BE DETACHED BY LIVE MIGRATION ──────────  (1)
%  VMS2460 LIVE MIGRATION OF VM (4,G4IVO) TO TARGET SU 'D021ZE01'
POSSIBLE ──────────────────────────────────────────────────────────────  (2)
```

The checks on the target SU have delivered the following results (after correcting the problems in the above example):

(1) Warnings: Alias devices are automatically detached and removed from the VM during VM Migration.

(2) The VM Migration is possible.

7.5.11 CREATE-VM (Initialize a VM)

/CREATE-VM creates and initializes a VM in VM2000.

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 (SU /390)
- (Member) CPU quota and maximum CPU utilization of the VM
- Maximum I/O utilization of the VM (SU /390)
- Adding of the VM to a VM group (SU /390)
- 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

  390 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 controlling the real CPU (SU /390)
- Monitor job variable for monitoring the VM
- PERSISTENT attribute

After the command has been executed successfully, the VM has state INIT-ONLY.

Depending on the PERSISTENT attribute,a persistent or non-persistent VM definition is configured for the VM.

x86 On SU x86 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.

i The monitor VM is automatically initialized when VM2000 is initialized. Its attributes and resources are configured when VM2000 is installed (see chapter "Installing VM2000").

This command is reserved for the VM2000 administrator.

CREATE-**VM**
MEMORY-SIZE = <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**) ,**MIN-MEM**ORY-**SIZE** = ***STD** / <integer 1..1048576>(**DIM**ENSION = ***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>

,**PASS**WORD = ***NONE** / <c-string 1..8> / <x-string 1..16>

,**PROCES**SOR = ***MONO** / ***BI** / ***TRIPLE** / ***QUADRO** / ***OCTO** / ***MAX**IMUM / ***EXTRA-AND-NORMAL** /

<integer 1..32>

,**LOCATION** = ***ANY** / ***BEFORE-VM**(...) / ***AFTER-VM**(...) / ***MEMORY-END** / <x-text 1..8>(...)

***BEFORE-VM**(...)

| **VM-ID**ENTIFICATION = <integer 2..99> / <name 1..8>

***AFTER-VM**(...)

| **VM-ID**ENTIFICATION = <integer 1..99> / <name 1..8>

<x-text 1..8>(...)

| **DIM**ENSION = ***BYTE** / ***MEGABYTE**

,**VM-INDEX** = ***ANY** / <integer 1..99>

,**ADMIN-PRIVIL**EGES = ***STD** / ***MIN**IMUM / ***MAX**IMUM

,**PRIVILEGES** = ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **IO-RESET** = ***NO** / ***Y**ES

| ,**IO-PRIORITY** = ***NO** / ***Y**ES

| ,**AUTO-SNAP-ASSIGN**MENT = ***Y**ES / ***NO**

| ,**ASSIGN-BY-GUEST** = ***NO** / ***Y**ES(...)

| ***Y**ES(...)

| | **SET-NAME** = ***STDSET** / ***ALL** / ***EXCEPT**(...) / list-poss(8): <name 1..8> / *STDSET

| | ***EXCEPT**(...)

| | | **SET-NAME** = list-poss(8): <name 1..8> / *STDSET

,VM-ACTIVE-IDLE= ***NO** / ***AT-DEDICATED-CPUS**

,MONJV= ***NONE** / <filename 1..54 without-gen-vers>

,**JV-PASS**WORD = ***NONE** / <c-string 1..4> / <x-string 1..8> / <integer -2147483647..2147483648>

,**MAX-MEM**ORY**-SIZE** = ***STD** / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**)

,**PERSISTENT** = ***NO** / ***YES**

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).

x86 On SU x86 a small amount of a VM's main memory is required for the firmware component. The main memory of a VM must be at least 1024 Mbytes in size. The value must be a multiple of 2 Mbytes.

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`.

x86 On SU x86 the main memory of a VM must be at least 1024 Mbytes in size. 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. The VM name must be unique on the Server Unit.

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`).

x86 See also the general conditions for SU x86 in paragraph "Recommendations for the definition and use of VM names" in section "Initializing a VM".

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 "Planning distribution of the CPU capacity to the VMs").

CPU-QUOTA = *BY-VM-GROUP(...)
The VM is added to a VM group at creation time (SU /390).

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 "Operating VM groups (SU /390)"). The default value is 1.

MAX-CPU-UTILIZATION = 100 / <fixed 0.01..100.00>
Maximum CPU utilization of the VM (see "Planning distribution of the CPU capacity to the VMs").

390 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 "Operating VM groups (SU /390)").

MAX-IO-UTILIZATION = 100 / <integer 1..100>
Maximum I/O utilization of the VM (SU /390, see "Use of IORM in VM2000 operation").

CPU-POOL-NAME =
Determines the CPU pool to which the VM is assigned. Every VM is always assigned to precisely one CPU pool.

x86 On SU x86 the CPU pool must possess at least one attached real CPU.

i 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 "Scheduling procedure").

CPU-POOL-NAME = *STD
When a (single) VM is initialized (CPU-QUOTA = *STD / <fixed 0.01..99.99>), it is assigned to the standard CPU pool.

390 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.

390 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 can be up to 8 bytes long.

PASSWORD = *NONE
No password is necessary.

PASSWORD = <c-string 1..8>
Password in character form.

PASSWORD = <x-string 1..16>
Password in hexadecimal form. The password comprising zeros only (X'00..00') is ignored.

PROCESSOR =
This defines the multiprocessor level of the VM (see "Virtual CPUs").

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 "Real CPUs".
390 On SU /390, the maximum number of virtual CPUs is 16.

PROCESSOR = *EXTRA-AND-NORMAL
A VM is configured with as many virtual CPUs as normal CPUs and extra CPUs are available for VM2000 operation.
390 On SU /390, up to 16 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 Unit 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 (SU /390).

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 digits). If necessary, the address is padded with leading zeros.

VM-INDEX =
The VM index.

VM-INDEX = *ANY
The VM index is assigned by VM2000.

VM-INDEX = <integer 1..99>
Specifies the VM index.
The largest possible VM index depends on the hardware. `/SHOW-VM-RESOURCES INFORMATION=*CONFIGURATION` provides the necessary information.

> If `VM-INDEX=*ANY` is specified and the VM migrated at a later point (VM Migration), the VM will receive any free VM index on the target SU. Otherwise, the specified (fixed) VM index must be available on the target SU as well.

ADMIN-PRIVILEGES =
Defines the command/function range for the VM2000 administrator and the VM administrator (see "Extending and restricting the range of commands/functions").

ADMIN-PRIVILEGES = *STD
Standard command/function range.

ADMIN-PRIVILEGES = *MINIMUM
Restricted command/function range for the VM2000 administrator.

390 The operand value `*MINIMUM` only be specified on SU /390 (`/CREATE-VM` in the configuration file of the monitor VM or when the monitor system is started up in dialog mode).

x86 On SU x86, 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.

390 This operand value is ignored in the configuration file for the monitor VM. `ADMIN-PRIVILEGES=*STD` is used.

PRIVILEGES = *PARAMETERS(...)
Describes privileges of the VM (see "Initializing a VM").

IO-RESET = *NO / *YES
Specifies on SU /390 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.

IO-PRIORITY = *NO / *YES
Specifies on SU /390 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.

> The total number of virtual CPUs of all the VMs with this privilege may not be greater than the number of the real normal CPUs of the Server Unit.

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 be assigned explicitly to the VM by the VM or VM2000 administrator.

SET-NAME =
Determines the assignment sets from which the guest system may implicitly assign devices to the VM. With /CREATE-VM, up to 8 assignment sets can be assigned to the VM by means of explicit list specification. Additional assignment sets (up to maximum of 28) can be assigned to the VM with /MODIFY-VM-ATTRIBUTES.

SET-NAME = *STDSET / list-poss (8): <name 1..8> / *STDSET
Explicit specification of the assignment sets.
The explicit list of the assignment sets can be extented up to 28 assignment sets with /MODIFY-VM-ATTRIBUTES.

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.
The explicit exception list of the assignment sets can be extented up to 28 assignment sets with /MODIFY-VM-ATTRIBUTES.

VM-ACTIVE-IDLE =
Determines on SU /390 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 "Initializing a VM".

VM-ACTIVE-IDLE = *NO
The VM leaves control over the real CPU if the VM's virtual CPU which runs on this becomes 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 is not restricted.

MONJV =
Defines a MONJV for monitoring the VM (see "Monitoring a VM with job variables (MONJV)").

MONJV = *NONE
No MONJV is defined.

MONJV = <filename 1..54 without-gen-vers>
The specified MONJV is supplied with data by VM2000; therefore, the sender of the command must 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 the specified file name does not refer to an existing job variable, 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 in section "Initializing a VM".

390 This operand has no effect for SU /390.
x86 The value must be a multiple of 2 Mbytes for SU x86.

PERSISTENT =
Determines whether a persistent VM is to be created.

PERSISTENT = *NO
A VM is created which is not persistent.
A non-persistent VM definition is created for the VM.

PERSISTENT = *YES
A persistent VM is created.
A persistent VM definition is created for the VM. The guest system on the VM is not started automatically (AUTO-IPL=*NO). The specifications for the automatic start (when the VM definition is activated with /ACTIVATE-VM-DEFINITION) can be modified with /MODIFY-VM-ATTRIBUTES after the VM has been initialized (see "MODIFY-VM-ATTRIBUTES (Modify attributes of a VM)").

Command return codes

(SC2)	SC1	Main code	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	VMS2305	VM definition could not be created
	64	VMS2310	Generation of the VM definition not confirmed in the expected time
	64	VMS2311	Maximum permitted number of VM definitions already reached
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	64	VMS3034	ADMIN-PRIVILEGES=*MIN only permissible for the monitor VM
	64	VMS3035	IO-PRIORITY cannot be set (total of virtual CPUs with privilege too high)
	64	VMS4000	VM not yet created (*BEFORE-VM or *AFTER-VM operand)
	64	VMS4010	Maximum permitted number of VMs already 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	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 (SU x86)

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           VERSION PER ADMIN PRIV
 2 SYSTEM2   INIT-ONLY               NO  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 0    4IVR    V0292021-02-09110022
 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) VM2000 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" and the detailed "Example of managing CPU pools".

7.5.12 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.

Devices can be assigned to a created assignment set using /MODIFY-VM-DEVICE-ATTRIBUTES. VMs can be privileged for this assignment set when the VM is initialized (see "Initializing a VM") or with /MODIFY-VM-ATTRIBUTES.

This command is reserved for the VM2000 administrator.

CREATE-**VM-ASSIGN**MENT-**SET**
SET-NAME = <name 1..8>

SET-NAME = <name 1..8>
Name of the new assignment set.

Command return codes

(SC2)	SC1	Main code	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 in the sections "Example of peripheral device management" and "Example of pubset management".

7.5.13 CREATE-VM-CPU-POOL (Create CPU pool)

`/CREATE-VM-CPU-POOL` is used to create a CPU pool in VM2000.

CPUs, VMs and VM groups (SU /390) can be assigned to a created CPU pool.

This command is reserved for the VM2000 administrator.

CREATE**-VM-CPU-POOL**
CPU-POOL-NAME = <name 1..8>

CPU-POOL-NAME = <name 1..8>
Name of the new CPU pool.
x86 On SU x86, the characters #, $ and @ should not be used in the CPU pool name.

Command return codes

(SC2)	SC1	Main code	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".

7.5.14 CREATE-VM-DEFINITION (Create VM definition)

/CREATE-VM-DEFINITION creates a persistent VM definition.

The following attributes and resources for the VM are entered in the VM definition:

- 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
- Multiprocessor level of the VM
- (Member) CPU quota and maximum CPU utilization of the VM
- Adding of the VM to a VM group (SU /390)
- Assignment of the VM to a CPU pool
- Settings for controlling the real CPU (SU /390)
- Maximum I/O utilization of the VM (SU /390)
- Devices for the VM
- The password for administration
- Range of commands for the VM administrator
- Privileges of the VM

390 As it is recommended that a VM be created without the privileges IO-PRIORITY and IO-RESET (see "CREATE-VM (Initialize a VM)"), IO-PRIORITY=*NO and IO-RESET=*NO should also be entered for the VM in the VM definition.

- Specifications for automatic startup of the guest system on the VM when activating the VM definition

After the command has been executed, a persistent VM definition with the specified VM name exists. Further /CREATE-VM or /CREATE-VM-DEFINITION commands with this VM name are rejected.

This command is reserved for the VM2000 administrator.

CREATE-**VM-DEF**INITION

VM-NAME = <name 1..8>

,**VM-INDEX** = ***ANY** / <integer 2..99>

,**MEMORY-SIZE** = <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**)

,**MIN-MEM**ORY-**SIZE** = ***STD** / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**)

,**MAX-MEM**ORY-**SIZE** = ***STD** / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**)

,**PROCES**SOR = **_MONO_** / ***BI** / ***TRIPLE** / ***QUADRO** / ***OCTO** / ***MAX**IMUM / ***EXTRA-AND-NORMAL** /

<integer 1..32>

,**CPU-QUOTA** = **_STD_** / <fixed 0.01..99.99>(...) / ***BY-VM-GROUP**(...)

<fixed 0.01..99.99>(...)

| **CPU-POOL-NAME** = **_*STDPOOL_** / <name 1..8>

***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>

,VM-ACTIVE-IDLE= **_*NO_** / ***AT-DEDICATED-CPUS**

,**MAX-IO-UTILIZATION** = **_100_** / <integer 1..100>

,DEVICES = **_*NONE_** / list-poss(28): ***PUB**SET**-DEV**ICES(...) / ***DEVICE-RANGE**(...) /

<alphanum-name 1..4 with-wild(20)(...)>

***PUB**SET**-DEV**ICES(...)

| **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| ***BY-PUBRES-DEV**ICE(...)

| | **UNIT** = <alphanum-name 2..2> / <x-text 4..4>

| ,**USAGE** = **_*EXCL_**USIVE / ***SHARED**

***DEVICE-RANGE**(...)

| **FROM** = <alphanum-name 2..2> / <x-text 4..4>

| ,**TO** = <alphanum-name 2..2> / <x-text 4..4>

| ,**USAGE** = **_*EXCL_**USIVE / ***SHARED**

<alphanum-name 1..4 with-wild(20)(...)>

| **USAGE** = **_*EXCL_**USIVE / ***SHARED** / ***VC**

,**PASS**WORD = ***NONE** / <c-string 1..8> / <x-string 1..16>

,**ADMIN-PRIVIL**EGES = ***STD** / ***MAX**IMUM

,**PRIVILEGES** = ***PAR**AMETERS(...)

***PAR**AMETERS(...)

| **IO-RESET** = ***NO** / ***Y**ES

| ,**IO-PRIORITY** = ***NO** / ***Y**ES

| ,**AUTO-SNAP-ASSIGN**MENT = ***Y**ES / ***NO**

| ,**ASSIGN-BY-GUEST** = ***NO** / ***Y**ES(...)

| ***Y**ES(...)

| | **SET-NAME** = ***STDSET** / ***ALL** / ***EXCEPT**(...) / list-poss(8): <name 1..8> / *STDSET

| | ***EXCEPT**(...)

| | | **SET-NAME** = list-poss(8): <name 1..8> / *STDSET

,AUTO-IPL= ***NO** / ***Y**ES(...)

***Y**ES(...)

| **IPL-UNIT** = <alphanum-name 2..2> / <x-text 4..4>

| ,**IPL-CONS**OLE = <alphanum-name 2..2>

| ,**MODE** = ***AUTO**MATIC / ***FAST**

| ,**SYSTEM-NAME** = ***NONE** / <name 1..8>

VM-NAME = <name 1..8>
Name of the VM definition.
Identical to the name of the virtual machine which is to be initialized.

x86 See also the general conditions for SU x86 in section “Recommendations for the definition and use of VM names” in section "Initializing a VM".

VM-INDEX =
The VM index.

VM-INDEX = *ANY
The VM index is assigned by VM2000 when the VM definition is activated.

VM-INDEX = <integer 2..99>
Specifies the VM index.

In the case of a standard name (e.g. VM0002), only the appropriate VM index (in this case 2) may be specified.

If VM-INDEX=*ANY is specified and the VM migrated at a later point (VM Migration), the VM will receive any free VM index on the target SU. Otherwise, the specified (fixed) VM index must be available on the target SU as well.

MEMORY-SIZE = <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)

Predefined size of main memory for the appropriate VM up to a size of 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

x86 On SU x86, a small amount of a VM's main memory is required for the firmware component. The main memory of a VM must be at least 1024 Mbytes large. The value must be a multiple of 2 Mbytes.

MIN-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)

Predefined 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.

x86 On SU x86, the main memory of a VM must be at least 1024 Mbytes in size. The value must be a multiple of 2 Mbytes.

MAX-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)

Predefined 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 in section "Initializing a VM".

390 This operand has no effect for SU /390.
x86 The value must be a multiple of 2 Mbytes for SU x86.

PROCESSOR =

Envisaged multiprocessor level of the VM (see "Virtual CPUs").

PROCESSOR = *MONO

The VM is to be created with one virtual CPU.

PROCESSOR = *BI

The VM is to be created with two virtual CPUs.

PROCESSOR = *TRIPLE

The VM is to be created with three virtual CPUs.

PROCESSOR = *QUADRO

The VM is to be created with four virtual CPUs.

PROCESSOR = *OCTO
The VM is to be created with eight virtual CPUs.

PROCESSOR = *MAXIMUM
A VM with the maximum possible number of virtual CPUs is to be created. The maximum possible number of virtual CPUs matches the number of real normal CPUs which are available for VM2000 operation, see "Real CPUs".
390 On SU /390, the maximum number of virtual CPUs is 16.

PROCESSOR = *EXTRA-AND-NORMAL
A VM is to be configured with as many virtual CPUs as normal CPUs and extra CPUs are available for VM2000 operation.
390 On SU /390, up to 16 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 Unit 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.

CPU-QUOTA =
Envisaged CPU quota of the VM.

CPU-QUOTA = *STD / <fixed 0.01..99.99>(...)
CPU quota of a VM which is not (yet) to be assigned to a VM group when the VM definition is activated. Determines the VM's 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 "Planning distribution of the CPU capacity to the VMs").
The default value is `1 (CPU-POOL-NAME=*STDPOOL)`, i.e. the VM is assigned to the standard CPU pool, but not to a VM group, when the VM definition is activated.

CPU-POOL-NAME =
Defines the CPU pool to which the VM is to be assigned when the VM definition is activated.
Every VM is always assigned to precisely one CPU pool.
When the VM definition is activated, the CPU pool must possess at least one attached real CPU.

CPU-POOL-NAME = *STDPOOL
The VM is assigned to the standard CPU pool.

CPU-POOL-NAME = <name 1..8>
Name of the CPU pool to which the VM is assigned.

CPU-QUOTA = *BY-VM-GROUP(...)
CPU quota of a VM which is to be assigned to the specified VM group when the VM definition is activated (SU /390). This determines the VM's CPU share in comparison to the other VMs in the VM group (see "Operating VM groups (SU /390)"). The VM is implicitly assigned to the CPU pool of the VM group.

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. The default value is 1.

MAX-CPU-UTILIZATION = 100 / <fixed 0.01..100.00>
Envisaged maximum CPU utilization of the VM (see "Planning distribution of the CPU capacity to the VMs").

390 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 "Operating VM groups (SU /390)").

VM-ACTIVE-IDLE =
Determines on SU /390 whether, in the event of fixed CPU assignment, 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 "Initializing a VM".

VM-ACTIVE-IDLE = *NO
The VM leaves control over the real CPU if the VM’s virtual CPU which runs on this becomes 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 "Initializing a VM") is not restricted.

MAX-IO-UTILIZATION = 100 / <integer 1..100>
Envisaged maximum I/O utilization of the VM (SU /390, see "Use of IORM in VM2000 operation").

DEVICES =
Determines the devices which are to be assigned to the VM when the VM definition is activated.

DEVICES = *NONE
No devices are to be assigned to the VM when the VM definition is activated.

DEVICES = *PUBSET-DEVICES(...)
The devices of a pubset are to be assigned to the VM when the VM definition is activated.

¡ See also "Notes on pubset specification in the VM2000 commands" and "Pubset reconfiguration".

PUBSET =
Determines the pubset. Only the pubset is noted in the VM definition. The devices of the pubset are determined only when the VM definition is activated.

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.

USAGE =
Specifies the usage mode which the devices of the VM's pubset are to be assigned when the VM definition is activated.

USAGE = *EXCLUSIVE
The devices are to be assigned exclusively to the VM.

USAGE = *SHARED
The devices are to be assigned to the VM as shareable (usage mode `SHARED`), see "Usage mode of a device".

DEVICES = *DEVICE-RANGE(...)

The VM is to be assigned all devices contained in this name range when the VM definition is activated. All devices currently contained in the name range are entered in the VM definition.

> ¡ 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.

390 If `UNITS` are specified with `*DEVICE-RANGE` or with wildcards, alias devices (PAV/DPAV) and CRD 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.

USAGE = *EXCLUSIVE / *SHARED
Specifies the usage mode which the devices in the name range of the VM are to be assigned when the VM definition is activated. See "operand USAGE for *PUBSET-DEVICES".

DEVICES = <alphanum-name 1..4 with-wild(20)>(...)
The VM is to be assigned the specified devices when the VM definition is activated. The devices are entered in the VM definition.

USAGE = *EXCLUSIVE / *SHARED
Specifies the usage mode which the devices of the VM are to be assigned when the VM definition is activated. See "operand USAGE for *PUBSET-DEVICES".

USAGE = *VC
The device is to be assigned exclusively to the VM as a virtual console when the VM definition is activated (SU /390). It must be appropriately generated. Only one device for each VM can be operated as a virtual console.

PASSWORD =
Password controlling access authorization in `/BEGIN-VM-DIALOG`. The password can be up to 8 bytes long.

PASSWORD = *NONE
No password is necessary.

PASSWORD = <c-string 1..8>
Password in character form.

PASSWORD = <x-string 1..16>
Password in hexadecimal form. The password comprising zeros only (X'00..00') is ignored.

ADMIN-PRIVILEGES =
Defines the envisaged command/function range for the VM administrator (see "Extending and restricting the range of commands/functions").

ADMIN-PRIVILEGES = *STD
Standard command/function range.

ADMIN-PRIVILEGES = *MAXIMUM
Extended command/function range for the VM administrator.

PRIVILEGES = *PARAMETERS(...)
Describes privileges of the VM (see "Initializing a VM").

IO-RESET = *NO / *YES
Specifies on SU /390 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.

IO-PRIORITY = *NO / *YES
Specifies on SU /390 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.

> The total number of virtual CPUs of all the VMs with this privilege may not be greater than the number of the real normal CPUs of the Server Unit.

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 be assigned explicitly to the VM by the VM or VM2000 administrator.

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.
The explicit list of the assignment sets can be extented up to 28 assignment sets with `/MODIFY-VM-DEFINITION`.

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. The explicit exception list of the assignment sets can be extented up to 28 assignment sets with /MODIFY-VM-DEFINITION.

AUTO-IPL =
Determines whether the guest system on the VM is to be started automatically when the VM definition is activated.

AUTO-IPL = *NO
The guest system on the VM is not started automatically.

AUTO-IPL = *YES(...)
The guest system on the VM is started automatically.
The parameters for the automatic startup must be appropriately supplied with values.

IPL-UNIT = <alphanum-name 2..2> / <x-text 4..4>
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.

IPL-CONSOLE = <alphanum-name 2..2>
Mnemonic device name of the KVP output device.

MODE = *AUTOMATIC / *FAST
Defines the type of system initialization (AUTOMATIC or FAST startup).

SYSTEM-NAME = *NONE / <name 1..8>
Specifies the system name.
390 The specification is ignored on SU /390.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	VMS4035	No corresponding device found (possibly only alias or CRD devices were specified)
1	0	VMS4180	Specified set of devices empty
	64	VMS2305	VM definition could not be created
	64	VMS2310	Generation of the VM definition not confirmed in the expected time
	64	VMS2311	Maximum permitted number of VM definitions already reached
	64	VMS2313	Device specified more than once with different usage modes
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	64	VMS3002	Invalid value for operand
	64	VMS4006	Device not available
	64	VMS4012	Requested memory size too small
	64	VMS4013	VM name not available
	64	VMS4016	Device cannot be used as a virtual console
	64	VMS4060	Only one virtual console possible per VM
	64	VMS4093	Invalid specifications for memory size of a VM
	64	VMS4190	Device is an alias device (PAV/DPAV)
	64	VMS4195	Device is a CRD device
	64	VMS4253	Pubres device of the pubset which was specified via the CATID cannot be determined
	130	VMS2202	The uniqueness of the VM name cannot currently be checked

Guaranteed messages

`VMS2301`

Example

See section "Examples of working with VM definitions".

7.5.15 CREATE-VM-GROUP (Create VM group)

x86 No VM groups are available on SU x86. 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.

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.

This command is reserved for the VM2000 administrator.

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 "Operating VM groups (SU /390)"). The default value is 1.

MAX-CPU-UTILIZATION = 100 / <fixed 0.01..100.00>
Maximum CPU utilization of the VM group, see "Operating VM groups (SU /390)". 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 "Scheduling procedure").

Command return codes

(SC2)	SC1	Main code	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".

7.5.16 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 VM definition of a non-persistent VM is deleted. The VM definition of a persistent VM is retained.
The monitor VM cannot be terminated by this command.

x86 On SU x86, the main memory is released asynchronously and this can take several seconds. Not till then, the released memory is 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 implicitly. The output of periodic monitored data (/SHOW-VM-STATUS PERIOD=<integer>) for the VM is also terminated.

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 = ***CURR**ENT / ***ALL** / <integer 1..99> / <name 1..8>
,CHECK-VM-STATE = ***Y**ES / ***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. 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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
2	0	VMS2306	VM definition could not be updated (SU /390, persistent VM)
2	0	VMS2308	VM definition could not be deleted (SU /390, non-persistent VM)
	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

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.

7.5.17 DELETE-VM-ASSIGNMENT-SET (Delete assignment set)

`/DELETE-VM-ASSIGNMENT-SET` is used to delete an assignment set.

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=...)` privilige).

The assignment set `*STDSET` cannot be deleted.

The EXCEPT list of the assignment sets for a VM will be reduced if required.

This command is reserved for the VM2000 administrator.

DELETE-**VM-ASSIGN**MENT-**SET**
SET-NAME = <name 1..8>

SET-NAME = <name 1..8>
Name of the assignment set.

Command return codes

(SC2)	SC1	Main code	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 ("Example of peripheral device management" and "Example of pubset management").

7.5.18 DELETE-VM-CPU-POOL (Delete CPU pool)

`/DELETE-VM-CPU-POOL` is used to delete a CPU pool.

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 `*STDPOOL` and the statistic CPU-Pool *POOL0 (SU x86) cannot be deleted.

This command is reserved for the VM2000 administrator.

DELETE**-VM-CPU-POOL**
CPU-POOL-NAME = <name 1..8>

CPU-POOL-NAME = <name 1..8>
Name of the CPU pool.

Command return codes

(SC2)	SC1	Main code	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".

7.5.19 DELETE-VM-DEFINITION (Delete VM definition)

`/DELETE-VM-DEFINITION` is used to delete a VM definition. The VM definition may not belong to an initialized VM.

This command is reserved for the VM2000 administrator.

DELETE-**VM-DEF**INITION
VM-NAME = <name 1..8>

VM-NAME = <name 1..8>
Name of the VM definition which is to be deleted.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
	64	VMS2300	VM definition does not exist
	64	VMS2309	VM definition could not be deleted
	64	VMS2310	Deletion of the VM definition not confirmed in the expected time
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	64	VMS4015	VM already initialized

Guaranteed messages

`VMS2302`.

Example

See section "Examples of working with VM definitions".

7.5.20 DELETE-VM-GROUP (Delete VM group)

x86 No VM groups are available on SU x86. The command is rejected with message VMS3011 and with general return code VMS9004.

/DELETE-VM-GROUP is used to delete a VM group.

A VM group can be deleted only if no VMs are still assigned to it.

This command is reserved for the VM2000 administrator.

DELETE**-VM-GROUP**
VM-GROUP-NAME = <name 1..8>

VM-GROUP-NAME = <name 1..8>
Name of the VM group.

Command return codes

(SC2)	SC1	Main code	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 '*STDPOOL'
VMS4405 VM GROUP 'GCUST1' DELETED
```

See also the detailed "Example of operating VM groups".

7.5.21 DETACH-VM-RESOURCES (Detach real CPUs)

`/DETACH-VM-RESOURCES` is used to detach attached real normal and extra CPUs for VM2000 operation.

This command is reserved for the VM2000 administrator.

DETACH-**VM-RES**OURCES

UNIT = ***CPU**(...) / ***EXTRA-CPU**(...)

***CPU**(...)

| **CPU-ID**ENTIFICATION = ***ANY**(...) / list-poss(63): <x-text 1..2>

| ***ANY**(...)

| | **CPU-POOL-NAME** = ***UNIQUE** / ***STDPOOL** / <name 1..8>

***EXTRA-CPU**(...)

| **CPU-ID**ENTIFICATION = ***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 cannot be detached.
>
> VM2000 prevents the last attached real CPU in a CPU pool from being detached in the following cases:
>
> 390 if a VM in RUNNING, IN HOLD (WAIT) or SETUP(MIGR) state is assigned to the CPU pool (message/return code `VMS4617`)
>
> x86 if a VM is still assigned to the CPU pool

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 "Increasing capacity with extra CPUs".

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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	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 M4IVR     *STDPOOL    20.00    1.00/ 12.04  80.00/ 80.00  2(+1)
 2 G4IVQ     *STDPOOL           26.00/ 15.66 100.00/100.00  4(+1)
 3 G4IVP     *STDPOOL    50.00    4.00/ 24.09 100.00/ 50.00  2(+1)
 4 G4IVO     *STDPOOL    50.00    1.00/  6.02 100.00/ 50.00  1(+1)
 5 G4IVJ     *STDPOOL           70.00/ 42.16 100.00/100.00  2(+1)
VM-ID        SCHED ACT-I  VIRT CPUS
 1 M4IVR      TS    NO    00(RUN),01(RUN),02(OFF)
 2 G4IVQ      TS    NO    00(RUN),01(RUN),02(STOP),03(STOP),
                          04(OFF)
 3 G4IVP      TS    NO    00(RUN),01(RUN),02(OFF)
 4 G4IVO      TS    NO    00(RUN),01(SLEEP)
 5 G4IVJ      -     NO    00(IDLE),01(IDLE),02(OFF)
CPU-ATTR     CPU-POOL     REAL CPUS
NORMAL       *STDPOOL     00(ATT),01(ATT)
EXTRA        *STDPOOL     02(ATT),03(ATT) ──────────────────────────────────  (2)
/detach-vm-resources unit=*extra-cpu ─────────────────────────────────────  (3)
VMS4077 EXTRA-CPU 02 DETACHED
VMS4077 EXTRA-CPU 03 DETACHED
/detach-vm-resources unit=*cpu(cpu-id=01) ─────────────────────────────────  (4)
VMS4045 CPU 01 DETACHED
/show-vm-resources information=*cpu ──────────────────────────────────────  (5)
VM-ID        CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC
 1 M4IVR     *STDPOOL    20.00    1.00/ 12.04  80.00/ 80.00  2(+1)
 2 G4IVQ     *STDPOOL           26.00/ 15.66 100.00/100.00  4(+1)
 3 G4IVP     *STDPOOL    50.00    4.00/ 24.09 100.00/ 50.00  2(+1)
 4 G4IVO     *STDPOOL    50.00    1.00/  6.02 100.00/ 50.00  1(+1)
 5 G4IVJ     *STDPOOL           70.00/ 42.16 100.00/100.00  2(+1)
VM-ID        SCHED ACT-I  VIRT CPUS
 1 M4IVR      TS    NO    00(RUN),01(RUN),02(OFF)
 2 G4IVQ      TS    NO    00(RUN),01(RUN),02(BLOCK),03(BLOCK),
                          04(OFF)
 3 G4IVP      TS    NO    00(RUN),01(RUN),02(OFF)
 4 G4IVO      TS    NO    00(RUN),01(SLEEP)
 5 G4IVJ      -     NO    00(IDLE),01(IDLE),02(OFF)
CPU-ATTR     CPU-POOL     REAL CPUS
NORMAL       *STDPOOL     00(ATT),01(DET)
EXTRA        *STDPOOL     02(DET),03(DET) ──────────────────────────────────  (6)
SPARE                     04(DET)
```

---

(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) Real normal CPU `01` is detached.

(5) The VM2000 administrator requests information on the CPU state.

(6) Real normal CPU `01` and the extra CPUs are detached.

See also the detailed "Example of managing CPU pools".

7.5.22 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 "Operating the guest systems using BS2000 consoles".

/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 = ***CURR**ENT / <integer 1..99> / <name 1..8>
,**DISCONNECT** = ***Y**ES / ***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 codes

(SC2)	SC1	Main code	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)

```
vm02:/end-vm-dialog ──────────────────────────────────────────────────────── (1)
VM01: %  VMS1012 PARTNER HAS TERMINATED AN ADMINISTRATION DIALOG WITH
         VIRTUAL MACHINE (2,SYSTEM2). ADMINISTRATION-ID USED '@002'────  (2)
VM01: %  VMS1032 PARTNER HAS TERMINATED A VC DIALOG WITH
         VIRTUAL MACHINE (2,SYSTEM2)
VM02:VMS3000 VM2000 COMMAND /END-VM-DIALOG FOR
     VIRTUAL MACHINE (2,SYSTEM2) PROCESSED SUCCESSFULLY
VM02:NBR0740 COMMAND COMPLETED 'END-VM-DIALOG';
     (RESULT: SC2=000, SC1=000, MC=CMD0001); DATE: <date>
@: OMS0006 CONNECTION TO PARTNER ($VMCONS/D021ZE01) LOST
   (PAC=VM02; 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.

7.5.23 EXTEND-VM-MEMORY (Increase main memory for a VM)

/EXTEND-VM-MEMORY enables 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.

After the main memory expansion, the new main memory sizes are noted in the VM definition.

The main memory of a VM cannot be enlarged while it is being migrated (VM Migration).

x86 On SU x86, the main memory can be extended to the maximum size set, see "Initializing a VM"and "Reconfiguring main memory".

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 state IN HOLD (WAIT).

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 (SU /390) 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.

x86 On SU x86, aborting a main memory reconfiguration with /START-VM, /RESET-VM or /DELETE-VM should be avoided.
Reason: The reconfiguration will only be aborted in BS2000 but continued in the firmware and the carrier system. A BS2000 IPL with /START-VM may be delayed until the reconfiguration in the carrier system is complete.

This command is reserved for the VM2000 administrator.

EXTEND-**VM-MEM**ORY
VM-IDENTIFICATION = <integer 1..99> / <name 1..8> ,**NEW-MEM**ORY-**SIZE** = ***UNCHA**NGED / ***MAX**IMUM / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**) ,**NEW-MIN-MEM**ORY-**SIZE** = ***UNCHA**NGED / ***MAX**IMUM / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**) **,MONJV**= ***NONE** / <filename 1..54 without-gen-vers> ,**JV-PASS**WORD = ***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).

390 If `*MAXIMUM` is specified, the main memory size is increased up to the next VM or to the end of the available main memory.

x86 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.

x86 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.

MONJV =
Defines a MONJV for monitoring the reconfiguration of the main memory (see the corresponding paragraph in section "Reconfiguring main memory").

MONJV = *NONE
No MONJV is defined.

MONJV = <filename 1..54 without-gen-vers>
The specified MONJV is supplied with data by VM2000; therefore, the sender of the command must 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 the specified file name does not refer to an existing job variable, 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.

i This operand is only evaluated if a previously created job variable was specified with `MONJV`.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS2070	VM in the 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
	128	VMS3042	VM being migrated

Guaranteed messages

VMS2306 (SU /390)

Example

```
/show-vm-resources information=*memory ──────────────────────────────────  (1)
VM-ID         LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR    00000000         16
 1 MONITOR    00000010        512        512
 2 SYSTEM2    00000210        512        512
NONE          00000410      14832
LARGEST FREE DOMAIN:       14832
/extend-vm-memory vm-id=2,new-memory-size=768 ──────────────────────────  (2)
/show-vm-resources information=*memory
VMS4094 MEMORY-RECONFIGURATION FOR VIRTUAL MACHINE (2,SYSTEM2) HAS BEEN
        SUCCESSFULLY PROCESSED. CURRENT MEMORY SIZES: 512 / 768 ─────────  (3)
VM-ID         LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR    00000000         16
 1 MONITOR    00000010        512        512
 2 SYSTEM2    00000210        768        512 ──────────────────────────────  (4)
NONE          00000510      14576
LARGEST FREE DOMAIN:       14576

------------------------------------------------------------------------------------------------------------
```

(1) The VM2000 administrator fetches information on the main memory distribution in VM2000. VM2 has a main memory size of 512 Mbytes and a minimum memory size of 512 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 768 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 768 Mbytes. This reduces the remaining free main memory.

7.5.24 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.

x86 On SU x86, a VM cannot be suspended. The command is rejected with the message VMS3011 and the general return code VMS9004.

The change of state is also noted in the VM definition.

A VM cannot be suspended while it is being migrated (VM Migration).

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 "Suspending a VM / resuming operation".

HOLD-VM
VM-IDENTIFICATION = ***CURR**ENT / ***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 running VMs except the monitor VM and the VM for which a VM Migration is running, 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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
2	0	VMS2306	VM definition could not be updated (SU /390)
	64	VMS1002	Command not permitted for the monitor VM
	64	VMS2101	VM2000 shutdown in progress
	64	VMS4000	VM not yet created
	64	VMS9004	Command not supported (SU x86)
	64	VMS9005	Command not or only partially processed
	128	VMS3042	VM being migrated

Guaranteed messages

VMS2000, VMS2012, VMS3041

Example

```
/show-vm-attributes vm-id=system2 ──────────────────────────────────────  (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)         512   (100)
VM-ID         STATE            VERSION PER ADMIN PRIV
 2 SYSTEM2    RUNNING          V20.0B  NO  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
              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)         512   (100)
VM-ID         STATE            VERSION PER ADMIN PRIV
 2 SYSTEM2    WAIT(SEL)        V20.0B  NO  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).

7.5.25 MIGRATE-VM (Migrate VM)

/MIGRATE-VM migrates the specified VM from the local Server Unit (source SU) to the required Server Unit (target SU).

After the synchronous checking of the prerequisites for the VM, control is given back to the caller. The VM Migration itself is executed asynchronously. It can be monitored with a MONJV.

During the VM Migration, the VM is only stopped for a short time. User operation, network transfers and transactions on the guest system continue normally on the target SU.

This command is reserved for the VM2000 administrator.

MIGRATE-**VM**
VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
,SERVER-UNIT-NAME = ***UNIQUE** / <composed-name 1..64>
,MONJV= ***NONE** / <filename 1..54 without-gen-vers>
,**JV-PASS**WORD = ***NONE** / <c-string 1..4> / <x-string 1..8> / <integer -2147483647..2147483648>

VM-IDENTIFICATION =
Identification of the virtual machine.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
Identifies the virtual machine by means of a VM index or VM name.

SERVER-UNIT-NAME =
Determines the Server Unit to which the VM is to be migrated (target SU). The target SU must belong to the same active SU cluster as the local SU (source SU).

SERVER-UNIT-NAME = *UNIQUE
If an SU cluster consists of only two Server Units, the remote SU is unique and chosen automatically as the target SU. Otherwise, the name of the target SU has to be specified.

SERVER-UNIT-NAME = <composed-name 1..64>
Name of the target SU.

390 In case of a target SU /390, this is the BS2000 host name of the monitor system.

MONJV =
Defines a MONJV for monitoring the VM Migration (see "Monitoring a VM Migration with job variables (MONJV)").

MONJV = *NONE
No MONJV is defined.

MONJV = <filename 1..54 without-gen-vers>
The specified MONJV is supplied with data by VM2000; therefore, the sender of the command must 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 the specified file name does not refer to an existing job variable, 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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required (source SU = target SU)
2	0	VMS2480	VM Migration initiated with warnings
	64	VMS2101	VM2000 shutdown in progress
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	64	VMS2341	No unique remote SU found
	64	VMS2400	Target SU not detected
	64	VMS2407	VM Migration rejected by guest system
	64	VMS2408	VM Migration not confirmed in time by the guest system
	64	VMS2462	Prerequisites for the VM Migration not met
	64	VMS4000	VM not yet created
	64	VMS4084	I/O configuration modification active
	64	VMS4340	MONJV could not be accessed

Guaranteed messages (on the source SU)

`VMS2401` through `VMS2407`, `VMS2409` through `VMS2426`, `VMS2440` through `VMS2444`, `VMS2450` through `VMS2452`, `VMS2470` through `VMS2472`, `VMS4024`, `VMS6021`

> In case of the messages `VMS2404` and `VMS2421` (migration rejected/aborted by the target SU), the cause is stated as a message insert in the form of the corresponding VM2000 message code.

Guaranteed messages (on the target SU)

VMS2050, VMS2430 through VMS2434, VMS3050, VMS6020

Example

See the example in the section "Example for a VM Migration (SU /390)".

7.5.26 MIGRATE-VM-DEFINITION (Migrate VM definition)

/MIGRATE-VM-DEFINITION migrates the specified VM definition from the local Server Unit (source SU) to the required Server Unit (target SU).

The VM definition may not belong to a VM which has already been initialized.

This command is reserved for the VM2000 administrator.

MIGRATE**-VM-DEF**INITION
VM-NAME = <name 1..8> **,SERVER-UNIT-NAME** = ***UNIQUE** / <composed-name 1..64>

VM-NAME = <name 1..8>
Identification of the VM definition by the VM name.

SERVER-UNIT-NAME =
Determines the Server Unit to which the VM definition is to be migrated (target SU). The target SU must belong to the same active SU cluster as the local SU (source SU).

SERVER-UNIT-NAME = *UNIQUE
If an SU cluster consists of only two Server Units, the remote SU is unique and chosen automatically as the target SU. Otherwise, the name of the target SU has to be specified.

SERVER-UNIT-NAME = <composed-name 1..64>
Name of the target SU.

390 In case of a target SU /390, this is the BS2000 host name of the monitor system.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS2321	Migration successful. However, the VM definition could not be deleted on the source SU.
	64	VMS2300	VM definition does not exist
	64	VMS2323	VM definition could not be migrated
	64	VMS2324	Migration of the VM definition rejected by the target SU
	64	VMS2325	Migration of the VM definition not confirmed in time by the target SU
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	64	VMS2341	No unique remote SU found
	64	VMS2400	Target SU not detected
	64	VMS2402	Target SU not available
	64	VMS2403	Connection to target SU aborted
	64	VMS2410	Target SU is not in the active SU cluster with the source SU
	64	VMS4015	VM already initialized

Guaranteed messages

VMS2320

Example

```
/migrate-vm-definition vm-name=g4ivn,server-unit-name=d020ze01
% VMS2320 DEFINITION OF VM 'G4IVN' MIGRATED TO TARGET SU 'D020ZE01'
```

7.5.27 MODIFY-VM-ATTRIBUTES (Modify attributes of a VM)

`/MODIFY-VM-ATTRIBUTES` is used to modify the attributes of an initialized VM.

The modified attributes and specifications are noted in the VM definition.

The following attributes can be modified:

- The CPU quota
- Maximum CPU utilization of the VM
- Maximum I/O utilization of the VM (SU /390)
- Password for administration
- Command range for VM2000 administrator and VM administrator
- The privileges for the VM
- Settings for controlling the real CPU (SU /390)
- The scope of the trace entries for VM2000 to be logged
- The PERSISTENT attribute
- The information for automatically starting the guest system on a persistent VM when next activating its VM definition

The attributes of a VM cannot be changed while it is being migrated (VM Migration).

> i You can modify the attributes and resources entered in the VM definition of a VM which has not yet been initialized using `/MODIFY-VM-DEFINITION`.

This command is reserved for the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM`.

MODIFY**-VM-ATTR**IBUTES

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>

,**CPU-QUOTA** = ***UNCHA**NGED / <fixed 0.01..99.99> / ***BY-VM-GROUP**(...)

***BY-VM-GROUP**(...)

| **MEMBER-CPU-QUOTA** = ***UNCHA**NGED / <fixed 0.01..99.99>

,**MAX-CPU-UTILIZATION** = ***UNCHA**NGED / <fixed 0.01..100.00>

,**MAX-IO-UTILIZATION** = ***UNCHA**NGED / <integer 1..100>

,**PASS**WORD =***UNCHA**NGED / ***NONE** / <c-string 1..8> / <x-string 1..16>

,**ADMIN-PRIVIL**EGES = ***UNCHA**NGED / ***STD** / ***MIN**IMUM / ***MAX**IMUM

,**PRIVILEGES** = ***PAR**AMETERS(...)

***PAR**AMETERS(...)

```
        |   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>
,PERSISTENT = *UNCHANGED / *NO / *YES(...)
    *YES(...)
        |   ,AUTO-IPL= *NO / *YES(...)
        |       *YES(...)
        |           |   IPL-UNIT = *UNCHANGED-OR-NONE / <alphanum-name 2..2> / <x-text 4..4>
        |           |   ,IPL-CONSOLE = *UNCHANGED-OR-NONE / <alphanum-name 2..2>
        |           |   ,MODE = *UNCHANGED-OR-AUTOMATIC / *AUTOMATIC / *FAST
        |           |   ,SYSTEM-NAME = *UNCHANGED-OR-NONE / *NONE / <name 1..8>
```

If no more detailed explanation is specified, the operand value *UNCHANGED means that the existing setting for the operand value is retained.

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 = <fixed 0.01..99.99>
New CPU quota for a VM which does not belong to a VM group, see "Planning distribution of the CPU capacity to the VMs".

CPU-QUOTA = *BY-VM-GROUP(...)
New member CPU quota for a VM which belongs to a VM group (SU /390), see "Operating VM groups (SU /390)".

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 = <fixed 0.01..100.00>
New maximum CPU utilization of the VM, see "Planning distribution of the CPU capacity to the VMs".

390 If the VM belongs to a VM group (SU /390), its maximum CPU utilization is also restricted by the "maximum CPU utilization of the VM group" (see "Operating VM groups (SU /390)").

MAX-IO-UTILIZATION =
On SU /390, determines whether the maximum I/O utilization of the VM is to be changed or retained.

MAX-IO-UTILIZATION = <integer 1..100>
New maximum I/O utilization of the VM, see "Use of IORM in VM2000 operation".

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 = *NONE
No password is to be defined.

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 "Extending and restricting the range of commands/functions").

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. This operand value may not be specified for the monitor VM.

PRIVILEGES = *PARAMETERS(...)
Describes privileges of the VM (see "Initializing a VM").
If `*UNCHANGED` is specified, the privileges of the VM remain unchanged.

IO-RESET = *NO / *YES
Specifies on SU /390 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.

IO-PRIORITY = *NO / *YES
Specifies on SU /390 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.

> The total number of virtual CPUs of all the VMs with this privilege may not be greater than the number of the real normal CPUs of the Server Unit.

AUTO-SNAP-ASSIGNMENT = *YES / *NO
Determines whether the guest system of the VM may assign snap units for Snapsets implicitly without the VM and devices being assigned the `ASSIGN-BY-GUEST` privilege or attribute.

ASSIGN-BY-GUEST = *NO / *YES(...)
Specifies whether devices can also be assigned implicitly to this VM by the guest system (`*YES`).
Alternatively (`*NO`), the devices must be assigned explicitly to the VM by the VM or VM2000 administrator.

SET-NAME =
Determines the assignment sets from which the guest system may assign devices implicitly to the VM. A VM can be 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 SU /390 whether, in the event of fixed CPU assignment (dedicated CPUs), a VM still retains control over the real CPU if the VM's virtual CPU which runs on this is inactive (interruptible wait state, "Idle"), see "Initializing a VM".

VM-ACTIVE-IDLE = *NO
The VM leaves control over the real CPU if the VM's virtual CPU which runs on this becomes 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 "Initializing a VM") is not restricted.

TRACE-LEVEL =
Specifies the scope of the trace entries for VM2000 to be logged (see "Other diagnostic aids"). This operand can only be specified for the monitor VM.
The default value after starting the monitor system 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 = <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 "Accounting in VM2000").

PERSISTENT =
Determines whether the PERSISTENT attribute is assigned to or removed from the VM which has already been initialized.
For a VM which is already persistent, the setting can be modified whether the guest system on the VM is to be started automatically the next time the VM definition is activated (AUTO-IPL operand). When the guest system on the VM which is already persistent is to be started automatically the next time the VM definition is activated, the specifications for the automatic startup of the guest system on the VM can be modified.
This operand cannot be specified for the monitor VM.

PERSISTENT = *NO
The PERSISTENT attribute is removed from a persistent VM. This is also entered in the VM definition. Specifications entered earlier for the IPL-UNIT, IPL-CONSOLE, MODE and SYSTEM-NAME parameters are deleted in the VM definition. The VM definition is no longer persistent.

PERSISTENT = *YES(...)
The VM is assigned or retains the PERSISTENT attribute. This is also entered in the VM definition. The VM definition becomes or remains persistent.

AUTO-IPL =
Determines whether the guest system on the VM is to be started automatically when the VM definition is activated.

AUTO-IPL = *NO
The guest system on the VM is not started automatically.

> The IPL-UNIT, IPL-CONSOLE, MODE and SYSTEM-NAME parameters set for a persistent VM are retained in the VM definition and can be reactivated using this command and the specification AUTO-IPL=*YES.

AUTO-IPL = *YES(...)
The guest system on the VM is started automatically.
The parameters for the automatic startup must be supplied with values appropriately.

IPL-UNIT = *UNCHANGED-OR-NONE
When a non-persistent VM is assigned the VM attribute PERSISTENT, IPL-UNIT=*NONE is entered in the VM definition. In the case of a VM which is already persistent (e.g. when switching from AUTO-IPL=*NO to AUTO-IPL=*YES), the existing entry for IPL-UNIT in the VM definition is retained.

IPL-UNIT = <alphanum-name 2..2> / <x-text 4..4>
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.

IPL-CONSOLE = *UNCHANGED-OR-NONE
When a non-persistent VM is assigned the VM attribute PERSISTENT, IPL-CONSOLE=*NONE is entered in the VM definition. In the case of a VM which is already persistent (e.g. when switching from AUTO-IPL=*NO to AUTO-IPL=*YES), the existing entry for IPL-CONSOLE in the VM definition is retained.

IPL-CONSOLE = <alphanum-name 2..2>
Mnemonic device name of the KVP output device.

MODE = *UNCHANGED-OR-AUTOMATIC

When a non-persistent VM is assigned the VM attribute PERSISTENT, `MODE=*AUTOMATIC` is entered in the VM definition. In the case of a VM which is already persistent (e.g. when switching from `AUTO-IPL=*NO` to `AUTO-IPL=*YES`), the existing entry for `MODE` in the VM definition is retained.

MODE = *AUTOMATIC / *FAST

Defines the type of system initialization (AUTOMATIC or FAST startup).

SYSTEM-NAME = *UNCHANGED-OR-NONE

When a non-persistent VM is assigned the VM attribute PERSISTENT, `SYSTEM-NAME=*NONE` is entered in the VM definition. In the case of a VM which is already persistent (e.g. when switching from `AUTO-IPL=*NO` to `AUTO-IPL=*YES`), the existing entry for `SYSTEM-NAME` in the VM definition is retained.

390 The specification is ignored on SU /390.

SYSTEM-NAME = *NONE / <name 1..8>

Specifies the system name.

390 The specification is ignored on SU /390.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
2	0	VMS2306	VM definition could not be updated (SU /390)
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	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 (total of virtual CPUs with privilege too high)
	64	VMS3038	PERSISTENT=*YES not permitted for the monitor VM
	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
	128	VMS3042	VM being migrated

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(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   RUNNING         V20.0B  NO  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.

7.5.28 MODIFY-VM-DEFINITION (Modify VM definition)

/MODIFY-VM-DEFINITION is used to modify a VM definition. The VM definition may not belong to a VM which has already been initialized.

The following attributes and resources can be modified in the VM definition:

- VM index
- Main memory size of the VM
- Minimum and maximum main memory size of the VM for main memory reconfiguration
- Multiprocessor level of the VM
- (Member) CPU quota and maximum CPU utilization of the VM
- Adding of the VM to a VM group (SU /390)
- Assignment of the VM to a CPU pool
- Settings for controlling the real CPU (SU /390)
- Maximum I/O utilization of the VM (SU /390)
- Devices of the VM
- The password for administration
- Range of commands for the VM administrator
- Privileges of the VM

  390 As it is recommended that a VM be created without the privileges IO-PRIORITY and IO-RESET (see "CREATE-VM (Initialize a VM)"), IO-PRIORITY=*NO and IO-RESET=*NO should also be entered for the VM in the VM definition.

- PERSISTENT attribute
- Specifications for automatic startup of the guest system on the VM when activating the VM definition

The VM name cannot be modified.

> ¡ The attributes of a VM which has already been initialized can be modified using /MODIFY-VM-ATTRIBUTES.

This command is reserved for the VM2000 administrator.

MODIFY-**VM-DEF**INITION
VM-NAME = <name 1..8> ,**VM-INDEX** = ***UNCHA**NGED / ***ANY** / <integer 2..99> ,**MEMORY-SIZE** = ***UNCHA**NGED / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**) ,**MIN-MEM**ORY-**SIZE** = ***UNCHA**NGED / ***STD** / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**) ,**MAX-MEM**ORY-**SIZE** = ***UNCHA**NGED / ***STD** /

<integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**)

,**PROCES**SOR = ***UNCHA**NGED / ***MONO** / ***BI** / ***TRIPLE** / ***QUADRO** / ***OCTO** / ***MAX**IMUM /

***EXTRA-AND-NORMAL** / <integer 1..32>

,**CPU-QUOTA** = ***UNCHA**NGED / <fixed 0.01..99.99>(...) / ***BY-VM-GROUP**(...)

<fixed 0.01..99.99>(...)

| **CPU-POOL-NAME** = ***UNCHANGED-OR-STDPOOL** / ***STDPOOL** / <name 1..8>

***BY-VM-GROUP**(...)

| **VM-GROUP-NAME** = <name 1..8>

| ,**MEMBER-CPU-QUOTA** = <fixed 0.01..99.99>

,**MAX-CPU-UTILIZATION** = ***UNCHA**NGED / <fixed 0.01..100.00>

,VM-ACTIVE-IDLE= ***UNCHA**NGED / ***NO** / ***AT-DEDICATED-CPUS**

,**MAX-IO-UTILIZATION** =***UNCHA**NGED / <integer 1..100>

,DEVICES = ***UNCHA**NGED / ***ADD**(...) / ***REM**OVE(...) / ***MOD**IFY(...)

***ADD**(...)

| **UNITS** = list-poss(28): ***PUB**SET**-DEV**ICES(...) / ***DEVICE-RANGE**(...) /

| <alphanum-name 1..4 with-wild(20)(...)>

| ***PUB**SET**-DEV**ICES(...)

| | **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| | ***BY-PUBRES-DEV**ICE(...)

| | | **UNIT** = <alphanum-name 2..2> / <x-text 4..4>

| | ,**USAGE** = ***EXCL**USIVE / ***SHARED**

| ***DEVICE-RANGE**(...)

| | **FROM** = <alphanum-name 2..2> / <x-text 4..4>

| | ,**TO** = <alphanum-name 2..2> / <x-text 4..4>

| | ,**USAGE** = ***EXCL**USIVE / ***SHARED**

| <alphanum-name 1..4 with-wild(20)(...)>

| | **USAGE** = ***EXCL**USIVE / ***SHARED** / ***VC**

*REMOVE(...)

| **UNITS** = ***ALL** / list-poss(28): ***PUB**SET**-DEV**ICES(...) / ***DEVICE-RANGE**(...) /

| <alphanum-name 1..4 with-wild(20)>

| ***PUB**SET**-DEV**ICES(...)

| | **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| | ***BY-PUBRES-DEV**ICE(...)

| | | **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>

*MODIFY(...)

| **UNITS** = list-poss(28): ***PUB**SET**-DEV**ICES(...) / ***DEVICE-RANGE**(...) /

| <alphanum-name 1..4 with-wild(20)(...)>

| ***PUB**SET**-DEV**ICES(...)

| | **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| | ***BY-PUBRES-DEV**ICE(...)

| | | **UNIT** = <alphanum-name 2..2> / <x-text 4..4>

| | ,**USAGE** = ***EXCL**USIVE / ***SHARED**

| ***DEVICE-RANGE**(...)

| | **FROM** = <alphanum-name 2..2> / <x-text 4..4>

| | ,**TO** = <alphanum-name 2..2> / <x-text 4..4>

| | ,**USAGE** = ***EXCL**USIVE / ***SHARED**

| <alphanum-name 1..4 with-wild(20)(...)>

| | **USAGE** = ***EXCL**USIVE / ***SHARED**

,**PASS**WORD = ***UNCHA**NGED / ***NONE** / <c-string 1..8> / <x-string 1..16>

,**ADMIN-PRIVIL**EGES = ***UNCHA**NGED / ***STD** / ***MAX**IMUM

```
,PRIVILEGES = *PARAMETERS(...)
    *PARAMETERS(...)
        | IO-RESET = *UNCHANGED / *NO / *YES
        | ,IO-PRIORITY = *UNCHANGED / *NO / *YES
        | ,AUTO-SNAP-ASSIGNMENT = *UNCHANGED / *YES / *NO
        | ,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
,PERSISTENT = *UNCHANGED / *YES(...)
    *YES(...)
        | ,AUTO-IPL= *NO / *YES(...)
        |     *YES(...)
        |         | IPL-UNIT = *UNCHANGED-OR-NONE / <alphanum-name 2..2> / <x-text 4..4>
        |         | ,IPL-CONSOLE = *UNCHANGED-OR-NONE / <alphanum-name 2..2>
        |         | ,MODE = *UNCHANGED-OR-AUTOMATIC / *AUTOMATIC / *FAST
        |         | ,SYSTEM-NAME = *UNCHANGED-OR-NONE / *NONE / <name 1..8>
```

If no more detailed explanation is specified, the operand value *UNCHANGED means that the existing setting for the operand value is retained.

VM-NAME = <name 1..8>
Name of the VM definition which is to be modified.

VM-INDEX =
The VM index.

VM-INDEX = *ANY
The VM index is assigned by VM2000 when the VM definition is activated.

VM-INDEX = <integer 2..99>
Specifies the VM index.

> In the case of a standard VM name (e.g. `VM0002`), only the appropriate VM index (in this case `2`) may be specified.
>
> If `VM-INDEX=*ANY` is specified and the VM migrated later on (VM Migration), the VM will receive any free VM index on the target SU. Otherwise, the specified (fixed) VM index must be available on the target SU as well.

MEMORY-SIZE = <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
Predefined size of main memory for the appropriate VM up to a size of 1 Tbyte (terabyte; 1 Tbyte = 1024 Gbytes = 1 048 576 Mbytes).

x86 On SU x86, a small amount of a VM's main memory is required for the firmware component. The main memory of a VM must be at least 1024 Mbytes large. The value must be a multiple of 2 Mbytes.

MIN-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
Predefined 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`.

x86 On SU x86, the main memory of a VM must be at least 1024 Mbytes in size. The value must be a multiple of 2 Mbytes.

MAX-MEMORY-SIZE = *STD / <integer 1..1048576>(DIMENSION = *MEGABYTE / *GIGABYTE)
Predefined 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 in section "Initializing a VM".

390 This operand has no effect for SU /390.

x86 The value must be a multiple of 2 Mbytes for SU x86.

PROCESSOR =
Envisaged multiprocessor level of the VM (see "Virtual CPUs").

PROCESSOR = *MONO
The VM is to be created with one virtual CPU.

PROCESSOR = *BI
The VM is to be created with two virtual CPUs.

PROCESSOR = *TRIPLE
The VM is to be created with three virtual CPUs.

PROCESSOR = *QUADRO
The VM is to be created with four virtual CPUs.

PROCESSOR = *OCTO
The VM is to be created with eight virtual CPUs.

PROCESSOR = *MAXIMUM
A VM with the maximum possible number of virtual CPUs is to be created. The maximum possible number of virtual CPUs matches the number of real normal CPUs which are available for VM2000 operation, see "Real CPUs".
390 On SU /390, the maximum number of virtual CPUs is 16.

PROCESSOR = *EXTRA-AND-NORMAL
A VM is to be configured with as many virtual CPUs as normal CPUs and extra CPUs are available for VM2000 operation.
390 On SU /390, up to 16 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 Unit 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.

CPU-QUOTA =
Envisaged CPU quota of the VM.

CPU-QUOTA = <fixed 0.01..99.99>(...)
CPU quota of a VM which is not (yet) to be assigned to a VM group when the VM definition is activated. Determines the VM's 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 "Planning distribution of the CPU capacity to the VMs").

CPU-POOL-NAME =
Defines the CPU pool to which the VM is to be assigned when the VM definition is activated.
Every VM is always assigned to precisely one CPU pool.
When the VM definition is activated, the CPU pool must possess at least one attached real CPU.

CPU-POOL-NAME = *UNCHANGED-OR-STDPOOL
If the VM was (previously) not to be assigned to a VM group, the existing assignment to a CPU pool is retained. If the VM was previously to be assigned to a VM group, it is no longer assigned to a VM group. It is assigned to the standard CPU pool.

CPU-POOL-NAME = *STDPOOL
The VM is assigned to the standard CPU pool.

CPU-POOL-NAME = <name 1..8>
Name of the CPU pool to which the VM is assigned.

CPU-QUOTA = *BY-VM-GROUP(...)
CPU quota of a VM which is to be assigned to the specified VM group when the VM definition is activated (SU /390). This determines the VM's CPU share in comparison to the other VMs in the VM group (see "Operating VM groups (SU /390)"). The VM is implicitly assigned to the CPU pool of the VM group.

VM-GROUP-NAME = <name 1..8>
Name of the VM group to which the VM is added.

MEMBER-CPU-QUOTA = <fixed 0.01..99.99>
Member CPU quota of the VM.

MAX-CPU-UTILIZATION = <fixed 0.01..100.00>
Envisaged maximum CPU utilization of the VM (see "Planning distribution of the CPU capacity to the VMs").

390 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 "Operating VM groups (SU /390)").

VM-ACTIVE-IDLE =
Determines on SU /390 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 "Initializing a VM".

VM-ACTIVE-IDLE = *NO
The VM leaves control over the real CPU if the VM's virtual CPU which runs on this becomes 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.

i With fixed CPU assignment, `VM-ACTIVE-IDLE=*AT-DEDICATED-CPUS` is only effective if the VM's maximum CPU utilization is not restricted.

MAX-IO-UTILIZATION = <integer 1..100>
Envisaged maximum I/O utilization of the VM (SU /390, see "Use of IORM in VM2000 operation").

DEVICES =
Changes the number of devices which is to be assigned to the VM when the VM definition is activated.

DEVICES = *ADD(...)
Further devices are to be assigned to the VM when the VM definition is activated.

UNITS =
Determines the devices which are to be assigned to the VM when the VM definition is activated.

UNITS = *PUBSET-DEVICES(...)
The devices of a pubset are to be assigned to the VM when the VM definition is activated.

> ¡ See also "Notes on pubset specification in the VM2000 commands" and "Pubset reconfiguration".

PUBSET =
Determines the pubset. Only the pubset is noted in the VM definition. The devices of the pubset are determined only when the VM definition is activated.

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.

USAGE =
Specifies the usage mode which the devices of the VM's pubset are to be assigned when the VM definition is activated.

USAGE = *EXCLUSIVE
The devices are to be assigned exclusively to the VM.

USAGE = *SHARED
The devices are to be assigned to the VM as shareable (usage mode `SHARED`), see "Usage mode of a device".

UNITS = *DEVICE-RANGE(...)

The VM is to be assigned all devices contained in this name range when the VM definition is activated. All devices currently contained in the name range are entered in the VM definition.

> ¡ 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.

390 If `UNITS` are specified with `*DEVICE-RANGE` or with wildcards, alias devices (PAV/DPAV) and CRD 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.

USAGE = *EXCLUSIVE / *SHARED
Specifies the usage mode which the devices in the name range of the VM are to be assigned when the VM definition is activated. See "operand USAGE for *PUBSET-DEVICES".

UNITS = <alphanum-name 1..4 with-wild(20)>(...)
The VM is to be assigned the specified devices when the VM definition is activated. The devices are entered in the VM definition.

USAGE = *EXCLUSIVE / *SHARED
Specifies the usage mode which the devices of the VM are to be assigned when the VM definition is activated. See "operand USAGE for *PUBSET-DEVICES".

USAGE = *VC
The device is to be assigned exclusively to the VM as a virtual console when the VM definition is activated (SU /390). It must be appropriately generated. Only one device for each VM can be operated as a virtual console.

DEVICES = *REMOVE(...)
Fewer devices are to be assigned to the VM when the VM definition is activated. The devices are removed from the VM definition.

ℹ Devices which are not contained in the VM definition are ignored.

UNITS =
Determines the devices which are no longer to be assigned to the VM when the VM definition is activated.

UNITS = *ALL
No device is to be assigned to the VM any more when the VM definition is activated.

UNITS = *PUBSET-DEVICES(...)
The devices of a pubset are no longer to be assigned to the VM when the VM definition is activated.

ℹ See also "Notes on pubset specification in the VM2000 commands" and "Pubset reconfiguration".

PUBSET =
Determines the pubset which is to be removed from the VM definition.

PUBSET = *BY-PUBRES-DEVICE(...)
Determines the pubset via 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(...)
The VM is no longer to be assigned the devices contained in this name range when the VM definition is activated. All devices currently contained in the name range are removed from the VM definition.

ℹ 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)>
The VM is no longer to be assigned the specified devices when the VM definition is activated. The devices are removed from the VM definition.

DEVICES = *MODIFY(...)
The usage mode of the devices entered in the VM definition is modified.

> ¡ Devices which are not contained in the VM definition are ignored.

UNITS =
Determines the devices whose usage mode is modified.

UNITS = *PUBSET-DEVICES(...)
The usage mode of the devices of a pubset is modified.

> ¡ See also "Notes on pubset specification in the VM2000 commands" and "Pubset reconfiguration".

PUBSET =
Determines the 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.

USAGE =
Specifies the usage mode which the devices of the VM's pubset are to be assigned when the VM definition is activated.

USAGE = *EXCLUSIVE
The devices are to be assigned exclusively to the VM.

USAGE = *SHARED
The devices are to be assigned to the VM as shareable (usage mode SHARED), see "Usage mode of a device".

UNITS = *DEVICE-RANGE(...)
The usage mode of all devices contained in this name range is modified.

> ¡ 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.

USAGE = *EXCLUSIVE / *SHARED
Specifies the usage mode which the devices in the name range of the VM are to be assigned when the VM definition is activated. See "operand USAGE for *PUBSET-DEVICES".

UNITS = <alphanum-name 1..4 with-wild(20)>(...)
The usage mode of the specified devices is modified.

USAGE = *EXCLUSIVE / *SHARED
Specifies the usage mode which the devices of the VM are to be assigned when the VM definition is activated. See "operand USAGE for *PUBSET-DEVICES".

PASSWORD =
Password controlling access authorization in `/BEGIN-VM-DIALOG`. The password can be up to 8 bytes long.

PASSWORD = *NONE
No password is necessary.

PASSWORD = <c-string 1..8>
Password in character form.

PASSWORD = <x-string 1..16>
Password in hexadecimal form. The password comprising zeros only (X'00..00') is ignored.

ADMIN-PRIVILEGES =
Defines the envisaged command/function range for the VM administrator (see "Extending and restricting the range of commands/functions").

ADMIN-PRIVILEGES = *STD
Standard command/function range.

ADMIN-PRIVILEGES = *MAXIMUM
Extended command/function range for the VM administrator.

PRIVILEGES = *PARAMETERS(...)
Describes privileges of the VM (see "Initializing a VM").

IO-RESET = *NO / *YES
Specifies on SU /390 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.

IO-PRIORITY = *NO / *YES
Specifies on SU /390 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.

> The total number of virtual CPUs of all the VMs with this privilege may not be greater than the number of the real normal CPUs of the Server Unit.

AUTO-SNAP-ASSIGNMENT = *YES / *NO
Determines whether the guest system on the VM may assign itself snap units for Snapsets implicitly without the VM and devices being assigned the ASSIGN-BY-GUEST privilege or attribute.

ASSIGN-BY-GUEST = *NO / *YES(...)
Specifies whether devices can also be assigned implicitly to this VM by the guest system (*YES).
Alternatively (*NO), the devices must be assigned explicitly to the VM by the VM or VM2000 administrator.

SET-NAME =
Determines the assignment sets from which the guest system may assign devices implicitly to the VM. A VM can be 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.

PERSISTENT = *YES(...)
Determines that the PERSISTENT attribute is to be assigned to the VM definition.
The VM definition becomes or remains persistent.
For a VM definition which is already persistent, the setting can be modified whether the guest system on the VM is to be started automatically the next time the VM definition is activated (AUTO-IPL operand). When the guest system on the VM which is already persistent is to be started automatically the next time the VM definition is activated, the specifications for the automatic startup of the guest system on the VM can be modified.

AUTO-IPL =
Determines whether the guest system on the VM is to be started automatically the next time the persistent VM definition is activated.

AUTO-IPL = *NO
The guest system on the VM is not started automatically.

i Specifications for the IPL-UNIT, IPL-CONSOLE, MODE and SYSTEM-NAME parameters made earlier are retained in the VM definition and can be reactivated using this command and the specification AUTO-IPL=*YES.

AUTO-IPL = *YES(...)
The guest system on the VM is started automatically.
The parameters for the automatic startup must be supplied with values appropriately.

IPL-UNIT = *UNCHANGED-OR-NONE
When a non-persistent VM definition is assigned the VM attribute PERSISTENT, IPL-UNIT=*NONE is entered in the VM definition. In the case of a VM definition which is already persistent (e.g. when switching from AUTO-IPL=*NO to AUTO-IPL=*YES), the existing entry for IPL-UNIT in the VM definition is retained.

IPL-UNIT = <alphanum-name 2..2> / <x-text 4..4>
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.

IPL-CONSOLE = *UNCHANGED-OR-NONE
When a non-persistent VM definition is assigned the VM attribute PERSISTENT, IPL-CONSOLE=*NONE is entered in the VM definition. In the case of a VM definition which is already persistent (e.g. when switching from AUTO-IPL=*NO to AUTO-IPL=*YES), the existing entry for IPL-CONSOLE in the VM definition is retained.

IPL-CONSOLE = <alphanum-name 2..2>
Mnemonic device name of the KVP output device.

MODE = *UNCHANGED-OR-AUTOMATIC
When a non-persistent VM definition is assigned the VM attribute PERSISTENT, MODE=*AUTOMATIC is entered in the VM definition. In the case of a VM definition which is already persistent (e.g. when switching from AUTO-IPL=*NO to AUTO-IPL=*YES), the existing entry for MODE in the VM definition is retained.

MODE = *AUTOMATIC / *FAST
Defines the type of system initialization (AUTOMATIC or FAST startup).

SYSTEM-NAME = *UNCHANGED-OR-NONE
When a non-persistent VM definition is assigned the VM attribute PERSISTENT, SYSTEM-NAME=*NONE is entered in the VM definition. In the case of a VM definition which is already persistent (e.g. when switching from AUTO-IPL=*NO to AUTO-IPL=*YES), the existing entry for SYSTEM-NAME in the VM definition is retained.
390 The specification is ignored on SU /390.

SYSTEM-NAME = *NONE / <name 1..8>
Specifies the system name.
390 The specification is ignored on SU /390.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	VMS4035	No corresponding device found (possibly only alias or CRD devices were specified)
1	0	VMS4180	Specified set of devices empty
	64	VMS2300	VM definition does not exist
	64	VMS2307	VM definition could not be modified
	64	VMS2310	Generation of the VM definition not confirmed in the expected time
	64	VMS2313	Device specified more than once with different usage modes
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
	64	VMS2315	Device already entered in VM definition with different usage mode
	64	VMS2317	Memory specifications do not match the values entered in the VM definition
	64	VMS3002	Invalid value for operand
	64	VMS4006	Device not available
	64	VMS4012	Requested memory size too small
	64	VMS4015	VM already initialized
	64	VMS4016	Device cannot be used as a virtual console
	64	VMS4060	Only one virtual console possible per VM
	64	VMS4093	Invalid specifications for memory size of a VM
	64	VMS4190	Device is an alias device (PAV/DPAV)
	64	VMS4195	Device is CRD
	64	VMS4211	Maximum number of assignment sets already assigned to the VM
	64	VMS4253	Pubres device of the pubset which was specified via the CATID cannot be determined

Example

See section "Examples of working with VM definitions".

7.5.29 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.

390 The default setting for this device attribute can be modified for SU /390 in the VM2000 parameter file (see "Setting the VM2000 parameters in the parameter service").

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 (see "Dynamic I/O configuration modification") then assume the new setting as a device attribute.

DUMMY tape devices should not be assigned implicitly, see "Special devices".

390 Virtual consoles cannot be assigned implicitly by a guest system. Alias devices (PAV/DPAV) and CRD devices may only be assigned implicitly (the attribute is ignored for these devices).

i The usage mode of a device does not change with this command (see "MODIFY-VM-DEVICE-USAGE (Modify usage mode of the devices of a VM)").

A message is only issued for a device in the event of an error.

This command is reserved 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" and "Pubset reconfiguration".

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 the attribute.

ASSIGN-BY-GUEST = *NO / *YES(...)
Specifies whether devices can also be assigned to this VM implicitly by the guest system (*YES). Alternatively (*NO), the devices must be assigned explicitly to the VM by the VM or VM2000 administrator. See "Initializing a VM".

ASSIGN-BY-GUEST = *NO
Device assignment may not be executed implicitly by the guest system, but only using the VM2000 command /ADD-VM-DEVICES or /SWITCH-VM-DEVICES or (for VM definitions) using /CREATE- or /MODIFY-VM-DEFINITION.
Exception: Alias devices (PAV/DPAV, see "Parallel access volume (PAV, SU /390)"), CRD devices (see "Special devices") and snap units for Snapsets (see "Special devices").

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 codes

(SC2)	SC1	Main code	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=vm04),
                               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 ("Example of peripheral device management" and "Example of pubset management").

7.5.30 MODIFY-VM-DEVICE-USAGE (Modify usage mode of devices of a VM)

/MODIFY-VM-DEVICE-USAGE changes the usage mode of VM devices, see "Usage mode of a device". 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 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, emulated devices and FastDPAV alias devices (SU /390)).

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)).

The change of usage mode is noted in the VM definition.

> This command does not change the attributes of the device (see "MODIFY-VM-DEVICE-ATTRIBUTES (Modify device attributes)").

390 Alias devices (PAV/DPAV) 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.

This command is reserved for the VM2000 administrator and VM administrators with ADMIN-PRIVILEGES=*MAXIMUM.

MODIFY**-VM-DEV**ICE**-USAGE**

UNITS = ***PUB**SET**-DEV**ICES(...) /

list-poss(28): ***DEVICE-RANGE**(...) / <alphanum-name 1..4 with-wild(20)>

***PUB**SET**-DEV**ICES(...)

| **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| ***BY-PUBRES-DEV**ICE(...)

| | **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-ID**ENTIFICATION = <integer 1..99> / <name 1..8>

,**USAGE** = ***EXCL**USIVE / ***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".

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 are specified with *DEVICE-RANGE or with wildcards, the devices that are not assigned to the VM are ignored.
>
> 390 On SU /390, alias devices (PAV/DPAV) and CRD devices are ignored additionally.

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 "Usage mode of a device". Shared devices are listed in the outputs of the VM2000 information commands under the keyword SHARED.

Command return codes

(SC2)	SC1	Main code	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	VMS2306	VM definition could not be updated (SU /390)
2	0	VMS4231	Usage mode not changed for 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/DPAV)
	64	VMS4195	Device is a CRD device
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command not or only partially processed
	128	VMS3042	VM being migrated

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, VMS4195

Example

```
/modify-vm-device-usage vm-id=system2,usage=*shared
                        units=*pubset-devices(pubset=vm04) ————————————  (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 VM04.

See also the detailed examples for device and pubset management ("Example of peripheral device management" and "Example of pubset management").

7.5.31 MODIFY-VM-GROUP-ATTRIBUTES (Modify attributes of a VM group)

x86 No VM groups are available on SU x86. 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 reserved for the VM2000 administrator.

MODIFY-**VM-GROUP-ATTR**IBUTES
VM-GROUP-NAME = <name 1..8>
,**CPU-QUOTA** = ***UNCHA**NGED / <fixed 0.01..99.99>
,**MAX-CPU-UTILIZATION** = ***UNCHA**NGED / <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 "Operating VM groups (SU /390)".

MAX-CPU-UTILIZATION = *UNCHANGED / <fixed 0.01..100.00>
Maximum CPU utilization of the VM group, see "Operating VM groups (SU /390)".

Command return codes

(SC2)	SC1	Main code	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".

7.5.32 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

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, the new passwords 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.

This command is reserved for the VM2000 administrator.

MODIFY**-VMCONS-PAR**AMETERS
OPERID-PASSWORD = ***UNCHA**NGED / ***NONE** / <c-string 1..8>
,**PASS**WORD =***UNCHA**NGED / ***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 codes

(SC2) SC1	Main code	Meaning
0	CMD0001	Command successfully executed

7.5.33 MOVE-VM (Move a VM within main memory)

x86 A VM need not be moved on SU x86. 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. A VM cannot be moved while it is being migrated (VM Migration).

This command is reserved for the VM2000 administrator.

MOVE-VM

LOCATION = ***BEFORE-VM**(...) / ***AFTER-VM**(...) / ***MEM**ORY-**END** / <x-text 1..8>(...)

***BEFORE-VM**(...)

| **VM-ID**ENTIFICATION = ***NEXT** / <integer 2..99> / <name 1..8>

***AFTER-VM**(...)

| **VM-ID**ENTIFICATION = ***PREVIOUS** / <integer 1..99> / <name 1..8>

<x-text 1..8>(...)

| **DIM**ENSION = ***BYTE** / ***MEGABYTE**

,**VM-ID**ENTIFICATION = <integer 2..99> / <name 1..8>

,**MSCF-COORDINATED** = ***IF-POSSIBLE** / ***Y**ES / ***NO**

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 VM (in the RUNNING state) is stopped by VM2000 (`WAIT(INT)` state, see "SHOW-VM-ATTRIBUTES (Output attributes of a VM)"), while it is moved in main memory. The duration of this process is not negligible for a large main memory of the VM (greater than 2 Gbyte; more than 10 seconds). For a guest system, 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 [10].

> This operand is ignored for a VM in the INIT-ONLY or DOWN status.

MSCF-COORDINATED = *IF-POSSIBLE
Protection is not requested in the guest system. 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 state cannot request the protection. The command is executed immediately.

MSCF-COORDINATED = *YES
Protection is not requested in the guest system. 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 state.

MSCF-COORDINATED = *NO
Protection is not requested.

Command return codes

(SC2)	SC1	Main code	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 protectionnot 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	VMS3042	VM being migrated
	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  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR   00000000         16
 1 MONITOR   00000010        512        512
 2 SYSTEM2   00000210        512        512
NONE         00000410      14832
LARGEST FREE DOMAIN:       14832
/move-vm location=*memory-end,vm-id=system2,mscf-coordinated=*if-poss  — (2)
/show-vm-resources information=*memory
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR   00000000         16
 1 MONITOR   00000010        512        512
NONE         00000210      14832
 2 SYSTEM2   00003C00        512        512
LARGEST FREE DOMAIN:       14832
/move-vm location=*after-vm(vm-id=monitor),vm-id=system2 ──────────────── (3)
/show-vm-resources information=*memory
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR   00000000         16
 1 MONITOR   00000010        512        512
 2 SYSTEM2   00000210        512        512
NONE         00000410      14832
LARGEST FREE DOMAIN:       14832

----------------------------------------------------------------------------------------------------------------
```

(1) The VM2000 administrator queries the distribution of main memory for 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.

7.5.34 RECOVER-VM-DEFINITION (Recover a VM definition)

`/RECOVER-VM-DEFINITION` recovers a VM definition, i.e. a VM definition is transferred from a remote Server Unit to the local Server Unit.

The remote Server Unit has to be in the same SU cluster as the local Server Unit. The remote Server Unit must not run. This check has to be done by the caller.

This command is reserved for the VM2000 administrator.

RECOVER**-VM-DEF**INITION
VM-NAME = <name 1..8> **,SERVER-UNIT-NAME = *UNIQUE** / <composed-name 1..64>

VM-NAME = <name 1..8>
Name of the VM definition to be recovered from the remote Server Unit.

SERVER-UNIT-NAME =
Name of the Server Unit from which the VM definition is to be recovered.

SERVER-UNIT-NAME = *UNIQUE
If an SU cluster consists of only two Server Units, the remote SU is unique and chosen automatically. Otherwise, the name of the remote SU has to be specified.

SERVER-UNIT-NAME = <composed-name 1..64>
Name of the remote Server Unit.
390 In case of a remote SU /390, this is the BS2000 host name of the monitor system.

Command return codes

(SC2) SC1	Main code	Meaning
0	CMD0001	Command successfully executed
64	VMS2300	VM definition does not exist
64	VMS2311	Maximum number of VM definitions already reached
64	VMS2332	Storage for VM definitions cannot be reached (SU /390)
64	VMS2341	No unique remote SU found
64	VMS2342	Specified SU is the local SU
64	VMS2343	Specified SU not in an SU cluster with the local SU
64	VMS2344	Remote SU is running
64	VMS2345	Unable to retrieve VM definition from the remote SU
64	VMS2346	VM definition is not unique in the SU cluster
64	VMS2347	Local SU not detected in any SU cluster
64	VMS2400	SU not detected
64	VMS4013	VM name not available

Guaranteed messages

`VMS2340`

Example

```
/recover-vm-definition vm-name=g4ivn
VMS2340 DEFINITION OF VM 'G4IVN' RECOVERED FROM REMOTE SU 'D021ZE01'
```

7.5.35 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.

After the main memory reduction, the new main memory size is noted in the VM definition.

The main memory of a VM cannot be reduced while the VM is being migrated (VM Migration).

Main memory reconfiguration is possible for VMs in states INIT-ONLY or DOWN. 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 state IN HOLD (WAIT).

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 (SU /390) 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.

x86 On SU x86, aborting a main memory reconfiguration with /START-VM, /RESET-VM or /DELETE-VM should be avoided.
Reason: The reconfiguration will only be aborted in BS2000 but continued in the firmware and the carrier system. A BS2000 IPL with /START-VM may be delayed until the reconfiguration in the carrier system is complete.

This command is reserved for the VM2000 administrator.

REDUCE**-VM-MEM**ORY
VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
,**NEW-MEM**ORY**-SIZE** = ***MIN**IMUM / <integer 1..1048576>(**DIM**ENSION = ***MEGABYTE** / ***GIGABYTE**)
,MONJV= ***NONE** / <filename 1..54 without-gen-vers>
,**JV-PASS**WORD = ***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).

390 On SU /390, the new main memory size may be below the MIN-MEMORY-SIZE of the VM concerned if the VM is in state INIT-ONLY or DOWN. In this case the value for the VM's MIN-MEMORY-SIZE is assigned the value of NEW-MEMORY-SIZE.

x86 On SU x86, 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 will be 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 the corresponding paragraph in section "Reconfiguring main memory").

MONJV = *NONE
No monitor job variable is defined.

MONJV = <filename 1..54 without-gen-vers>
The specified MONJV is supplied with data by VM2000; therefore, the sender of the command must 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 the specified file name does not refer to an existing job variable, 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.

i This operand is only evaluated if a previously created monitor job variable was specified with MONJV.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
	64	VMS2070	VM in the 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
	128	VMS3042	VM being migrated
	128	VMS4092	Memory reconfiguration is still active

Guaranteed messages

VMS2306 (SU /390)

Example

```
/reduce-vm-memory vm-id=2,new-memory-size=*minimum ————————————————————  (1)
/show-vm-resources information=*memory ———————————————————————————————  (2)
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR   00000000         16
 1 MONITOR   00000010        512        512
 2 SYSTEM2   00000210        768        512                   512
NONE         00000510      14576
LARGEST FREE DOMAIN:       14576
/show-vm-resources information=*memory
VMS4094 MEMORY-RECONFIGURATION FOR VIRTUAL MACHINE (2,SYSTEM2) HAS BEEN
        SUCCESSFULLY PROCESSED. CURRENT MEMORY SIZES: 512 / 512 ————————  (3)
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR   00000000         16
 1 MONITOR   00000010        512        512
 2 SYSTEM2   00000210        512        512 ————————————————————————————  (4)
NONE         00000410      14832
LARGEST FREE DOMAIN:       14832

------------------------------------------------------------------------------------------------------------
```

(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 768 Mbytes and a minimum memory size of 512 Mbytes. The planned new memory size of 512 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 512 Mbytes (again). This increases the remaining free main memory.

7.5.36 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 deallocated devices are removed from the VM definition.

> The devices removed from a VM should be disconnected beforehand in the corresponding guest system with /DETACH-DEVICE.

390 Alias devices (PAV/DPAV) and CRD devices cannot be explicitly removed.

REMOVE**-VM-DEV**ICES

UNITS = ***ALL**(...) / ***PUB**SET**-DEV**ICES(...) /

list-poss(28): ***DEVICE-RANGE**(...) / <alphanum-name 1..4 with-wild(20)>

***ALL**(...)

| **CHECK-VM-STATE** = ***Y**ES / ***NO**

***PUB**SET**-DEV**ICES(...)

| **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| ***BY-PUBRES-DEV**ICE(...)

| | **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-ID**ENTIFICATION = ***CURR**ENT / <integer 1..99> / <name 1..8>

,**FORCE** = ***NO**(...) / ***Y**ES

***NO**(...)

| **WAIT-FOR-UNASSIGN** = ***NO** / ***Y**ES

UNITS =
Defines the devices to be removed from the VM.

UNITS = *ALL(...)
All the devices assigned to this VM are removed. Alias devices (PAV) and CRD 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".

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` are specified with `*DEVICE-RANGE` or with wildcards, the devices that are not assigned to the VM are ignored.
>
> 390 On SU /390, alias devices (PAV/DPAV) and CRD devices are ignored additionally.

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 which are to be removed from the VM.

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.
No devices can be removed from a VM with FORCE=*NO while it is being migrated (VM 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 implicitly to the VM until the BS2000 command /DETACH-DEVICE releases this device implicitly 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.
390 On SU /390 the IO-RESET privilege of the VM must be noted (see "Initializing a VM").

Command return codes

(SC2)	SC1	Main code	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 assigned implicitly to the VM until it is detached in the guest system (VMS4087)
2	0	VMS2306	VM definition could not be updated (SU /390)
2	0	VMS4226	Not all of the pubset's devices could be removed
2	0	VMS4235	Some devices of the pubset remain assigned implicitly 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/DPAV)
	64	VMS4195	Device is a CRD device
	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
	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, VMS4195`

Example

```
/remove-vm-devices vm-id=system2,units=*pubset-devices(pubset=4ivl) ——   (1)
VMS4085 DEVICE 'FC1C' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
VMS4085 DEVICE 'FC1D' STILL ATTACHED IN GUEST SYSTEM. DEVICE NOT REMOVED
VMS4223 VM2000 COMMAND /REMOVE-VM-DEVICES FOR PUBSET SPECIFIED BY PUBRES
        DEVICE OR CATID '4IVL' REJECTED ————————————————————————————   (2)
/remove-vm-devices units=*pubset-devices(pubset=4ivl),vm-id=system2,
                   force=*no(wait-for-unassign=*yes) ————————————————   (3)
VMS4087 DEVICE 'FC1C' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN
        GUEST SYSTEM HAS BEEN DONE
VMS4087 DEVICE 'FC1D' REMAINS ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN
        GUEST SYSTEM HAS BEEN DONE
VMS4235 SOME DEVICES OF PUBSET '4IVL' WITH PUBRES DEVICE 'FC1C' REMAIN
        ASSIGNED TO VM (2,SYSTEM2) UNTIL DETACH IN GUEST SYSTEM HAS BEEN DONE
/remove-vm-devices units=(ch,ci,990f),vm-id=system2, ————————————————   (4)
VMS4009 DEVICE 'CH' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE 'CI' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)
VMS4009 DEVICE '990F' REMOVED FROM VIRTUAL MACHINE (2,SYSTEM2)

------------------------------------------------------------------------------------------------------------
```

(1) The VM2000 administrator wants to remove the pubset devices of pubset `4IVL` from VM2.

(2) The devices cannot be removed because they are still attached in the guest system.

(3) 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).

(4) Further devices are removed from VM2.

See also the detailed examples for device and pubset management ("Example of peripheral device management" and "Example of pubset management").

7.5.37 REMOVE-VM-FROM-VM-GROUP (Remove VM from VM group)

x86 No VM groups are available on SU x86. 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.

The change of assignment is noted in the VM definition.

A VM cannot be removed from a VM group while it is being migrated (VM Migration).

This command is reserved 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 "Planning distribution of the CPU capacity to the VMs"). The VM's previous member CPU quota of the VM is relinquished.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Decimal specifications rounded to 2 decimal places
2	0	VMS2306	VM definition could not be updated (SU /390)
	64	VMS4000	VM not yet created
	64	VMS4427	VM does not belong to VM group
	128	VMS3042	VM being migrated

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".

7.5.38 RESET-VM (Abort 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 change of state is noted in the VM definition.

The attributes and resources of the VM remain intact. ADMIN and VC dialogs continue to run. Implicitly assigned devices (see "Assignment sets, implicit device assignment and release") are, however, removed from the VM. If main memory reconfiguration is being carried out, it is terminated.

The guest system on a VM cannot be aborted while the VM is being migrated (VM Migration).

390 On SU /390, the Channel Subsystem Monitoring Facility function is switched off for the guest system.

The following tasks are facilitated with /RESET-VM:

- Final termination of a guest system which seems to have failed
- Immediate termination of a test system if shutdown is not wished for or not possible

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 by mistake.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS2306	VM definition could not be updated (SU /390)
	64	VMS3036	VM name and VM index do not correspond
	64	VMS4000	VM not yet created
	128	VMS3042	VM being migrated

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(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   RUNNING         V20.0B  NO  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(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   INIT-ONLY               NO  NO    AS ————————————————————  (3)
```

---

(1) The guest system on VM2 which is still active but defective is aborted.

(2) Asynchronous message about the guest system state.

(3) The new state of the VM is INIT-ONLY. The VM retains its attributes and resources (with the exception of implicitly assigned devices).

7.5.39 RESTORE-VM-CONFIGURATION (Restoring the VM2000 configuration)

`/RESTORE-VM-CONFIGURATION` is used to restore the VM2000 configuration from the file in which it was was previously backed up. See section "Backing up and restoring the VM2000 configuration".

> ¡ Persistent VMs are not contained in the backup file. They can be activated again with `/ACTIVATE-VM-DEFINITION` after the VM2000 configuration has been restored.

This command is reserved for the VM2000 administrator.

RESTORE**-VM-CONF**IGURATION
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 codes

(SC2)	SC1	Main code	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
VMS4801 VM2000 CONFIGURATION RESTORED FROM FILE ':4IVZ:$TSOS.VM2000-CONF'
```

7.5.40 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.

x86 On SU x86, a VM can be neither suspended nor resumed. The command is rejected with the message VMS3011 and the general return code VMS9004.

The change of state is noted in the VM definition.

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 VM 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 "Suspending a VM / resuming operation".

RESUME**-VM**
VM-IDENTIFICATION = ***CURR**ENT / ***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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required
2	0	VMS2020	VM remains in state IN HOLD(WAIT)
2	0	VMS2306	VM definition could not be updated (SU /390)
	64	VMS2001	VM not in state IN HOLD(WAIT)
	64	VMS4000	VM not yet created
	64	VMS4612	No attached CPU in the CPU pool
	64	VMS9004	Command not supported (SU x86)
	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)
             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)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   WAIT(SEL)       V20.0B  NO  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
             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)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   RUNNING         V20.0B  NO  NO    AS ──────────────────────  (4)

--------------------------------------------------------------------------------------------------------------
```

(1) The VM2000 administrator requests the state 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.

7.5.41 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".

> Persistent VMs are not saved. For each persistent VM a warning (`VMS4817`) is issued. They can also be restored with `/ACTIVATE-VM-DEFINITION` after the VM2000 configuration has been restored with `/RESTORE-VM-CONFIGURATION`.

The VM2000 configuration cannot be saved while a VM is being migrated (VM Migration).

This command is reserved for the VM2000 administrator.

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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS4814	An extra CPU is attached when the VM2000 configuration is restored from the backup file
2	0	VMS4818	Warnings occur when saving the VM2000 configuration
	64	VMS2070	VM in the 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
	128	VMS4830	VM has not yet reached SYSTEM READY
	128	VMS4831	Memory reconfiguration being processed
	128	VMS4832	Device configuration during processing

Guaranteed messages

`VMS4800, VMS4817`

Example

```
/save-vm-configuration to-file=vm2000-conf
VMS4800 CURRENT VM2000 CONFIGURATION SAVED IN FILE ':4IVZ:$TSOS.VM2000-CONF'
```

7.5.42 SEND-VM-MESSAGE (Pass messages to VM administrators)

/SEND-VM-MESSAGE enables messages to be sent to the VM administrators.

SEND-**VM-MES**SAGE
MESSAGE = <c-string 1..72> ,**VM-ID**ENTIFICATION = ***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 codes

(SC2)	SC1	Main code	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.

(6) This message appears in the ADMIN dialog for VM2.

When administration is carried out in a user task, message `VMS2036` is displayed as an asynchronous message in the user event stream or at the next input.

7.5.43 SHOW-VM-ASSIGNMENT-SET (Output assignment sets)

`/SHOW-VM-ASSIGNMENT-SET` is used to output the attributes of assignment sets.

The command provides the following information:

- Name of the assignment set
- VMs which may assign devices from the relevant assignment set implicitly
- Number of devices which are assigned to the assignment set

This command is reserved for the VM2000 administrator.

SHOW-**VM-ASSIGN**MENT-**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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
	64	VMS4200	Assignment set not yet created

Output in S variables (in user task)

Output information	Name of S variable	T	Contents
Name of the assignment set	var(*LIST).ASSIGNMENT-SET	S	*STDSET <name 1..8>
Index of the VM which may assign devices from this assignment set implicitly	var(*LIST).VM(*LIST).VM-INDEX	I	<integer 1..99>

Output information	Name of S variable	T	Contents
Name of the VM which may assign devices from this assignment set implicitly	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 assign devices from this assignment set implicitly by means of the VM index and VM name.

`NONE` No VM may assign devices from this assignment set implicitly.

Example

See also the detailed examples for device and pubset management ("Example of peripheral device management" and "Example of pubset management").

7.5.44 SHOW-VM-ATTRIBUTES (Output attributes of a VM)

`/SHOW-VM-ATTRIBUTES` is used to output the attributes of initialized virtual machines.

The command provides the following information:

- The attributes of the VM
- The VM state
- The version of the BS2000 operating system in 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 (SU /390)
- The CPU pool to which the VM is assigned
- The scheduling procedure of the VM
- The settings of the VM for control via the real CPU (SU /390)
- The state of the virtual CPUs
- The devices assigned to the VM
- Start address and size of the VM's main memory (SU /390)
- The monitor job variables assigned to the VM

> The information on the attributes and resources contained in the VM definition can be obtained using `/SHOW-VM-DEFINITION`.

SHOW-**VM-ATT**RIBUTES

INFORMATION = ***STD** / ***CPU** / ***UNITS** / ***MEMORY** / ***MONJV** / ***PRIVILEGES** / ***ALL**

,**TYPE** = ***ALL** / ***SD** / ***VC** / <family> / <device>

,**VM-ID**ENTIFICATION = ***CURR**ENT / ***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 VM attributes and states.

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 = *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`, `*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>	**device family (group)**
*CONSOLE	Console
*PRINTER	Printer
*FAM50	Special devices
*TD	TRANSDATA device
*UR	Unit record device
*DISK	Disk 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=?`. The device types on the current system can be displayed by means of `/SHOW-DEVICE-CONFIGURATION`, for example.

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 only permitted 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 may 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 (SU /390).
This operand is only permitted for the VM2000 administrator.

VM-GROUP-NAME = <name 1..8>
Name of the VM group.

VM-IDENTIFICATION = *BY-CPU-POOL(...)
No selection is made on this basis.
This operand is only permitted for the VM2000 administrator.

CPU-POOL-NAME = *STDPOOL / <name 1..8>
Name of the CPU pool.

Command return codes

(SC2)	SC1	Main code	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 is used to specify which S variables are to be supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
INFORMATION = *STD	1
INFORMATION = *CPU	2
INFORMATION = *UNITS	3
INFORMATION = *MEMORY	4
INFORMATION = *MONJV	6
INFORMATION = *PRIVILEGES	7
INFORMATION = *ALL	8

Output information	Name of S variable	T	Contents	Cond.
The VM index	var(*LIST).VM-INDEX	I	<integer 1..99>	1 - 8
Name of VM	var(*LIST).VM-NAME	S	<name 1..8>	1 - 8
Multiprocessor level of the VM	var(*LIST).NUM-OF-CPUS	I	<integer 1..32>	1,2,8
Number of virtual spare CPUs of the VM	var(*LIST).NUM-OF-VM-SPARE-CPUS	I	<integer 0..15>	1,2,8
Name of the CPU pool	var(*LIST).CPU-POOL	S	*STDPOOL <name 1..8>	1,2,8
CPU quota of the VM (integer)	var(*LIST).CPU-QUOTA	I	<integer 1..99>	1,2,8

Output information	Name of S variable	T	Contents	Cond.
CPU quota of the VM (exact)	var(*LIST).FINE-CPU-QUOTA	S	<fixed 0.01..99.99>	1,2,8
Effective CPU share of the VM	var(*LIST).EFFECTIVE-CPU-QUOTA	S	<fixed 0.01..100.00>	1,2,8
Maximum CPU utilization of the VM (integer)	var(*LIST).MAX-CPU-UTILIZATION	I	<integer 1..100>	1,2,8
Maximum CPU utilization of the VM (exact)	var(*LIST).FINE-MAX-UTIL	S	<fixed 0.01..100.00>	1,2,8
Maximum CPU utilization of the VM (effective)	var(*LIST).EFFECTIVE-MAX-UTIL	S	<fixed 0.01..100.00>	1,2,8
Name of the VM group	var(*LIST).VM-GROUP	S	*NONE <name 1..8>	1,2,8
Size of VM main memory in Mbytes	var(*LIST).MEM-SIZE	I	<integer 1..1048576>	1,4,8
Minimum size of VM main memory in Mbytes	var(*LIST).MIN-MEM-SIZE	S	*UNDEF <integer 1..1048576>	1,4,8
New main memory size for VM for main memory reconfiguration in Mbytes	var(*LIST).PLANNED-MEM-SIZE	S	*UNDEF <integer 1..1048576>	1,4,8
Is the IO-RESET privilege assigned?	var(*LIST).IO-PRIVIL	B	TRUE / FALSE	1,7,8
Is the IO-PRIORITY privilege 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

Output information	Name of S variable	T	Contents	Cond.
Status of VM	var(*LIST).STA	S	*INIT-ONLY / *RUN / *WAIT / *DOWN / *SETUP	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 global /HOLD-VM from the VM 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	BS2000 *UNKNOWN	1,8
Version of the guest operating systems	var(*LIST).GUEST-VERSION	S	<c-string 1..8> *UNKNOWN	1.8
Does the VM have the PERSISTENT attribute?	var(*LIST).PERSISTENT	B	TRUE / FALSE	1.8
Is the VM currently being migrated?	var(*LIST).MIGRATING	B	TRUE / FALSE	1,8
Does the VM have a different I/O configuration after the migration	var(*LIST).DIFFERENT-IO-CONF	B	TRUE / FALSE	1,8
Scheduling procedure	var(*LIST).SCHEDULE	S	*NONE *BY-TIMESLICE *DEDICATED-CPUS	2,8
CPU quota (exact) of the VM group if the VM is assigned to a VM group	var(*LIST).VM-GROUP-CPU-QUOTA	S	*NONE <fixed 0.01..99.99>	2,8

Output information	Name of S variable	T	Contents	Cond.
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
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 in Mbytes	var(*LIST).BS2-MEM-SIZE	I	<integer 1..1048576>	4,8
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
Monitor job variable for monitoring the migration for the VM	var(*LIST).MIGR-MONJV	S	<name 1..54> / NONE	6,8

Output information	Name of S variable	T	Contents	Cond.
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	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  1(+1)       512  (100)
VM-ID         STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2    RUNNING         V20.0B  NO  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` (SU /390).

`<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 "Use of IORM in VM2000 operation".

STATE

State of the VM.
If there are overlapping wait states, only the wait state with the highest priority (in the order SEL, GLB, VMA, see "Suspending a VM/resuming operation") is displayed.

INIT-ONLY	The VM is initialized (after VM initialization, after /RESET-VM or /ACTIVATE-VM-DEFINITION AUTO-IPL=*NO), but no guest system has been started yet
RUNNING	The guest system on the VM has been started and is running
WAIT(...)	The VM is in the wait state (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)

A VM that is currently being migrated has the following states:

- on the source SU:
  the current state of the VM, complemented by (MIGR), e.g. RUNNING(MIGR)
- on the target SU (SU /390):
  state SETUP(MIGR)

If the I/O configuration on the target SU is different from the I/O configuration on the source SU, the current state of the migrated VM is output on the target SU, complemented by (DIFF), e.g. RUNNING(DIFF)

VERSION

Version of the BS2000 operating system in the guest system in the form Vnn.na. If the version is not (yet) known, blanks are displayed.

PER

Displays whether the VM has the PERSISTENT attribute (YES/NO).

ADMIN

Indicates if a VM administrator is connected to the virtual machine (YES/NO).

PRIV

Privileges of the VM:

```
 AG    ASSIGN-BY-GUEST

 AS    AUTO-SNAP-ASSIGNMENT

 IO-P  IO-PRIORITY

 IO-R  IO-RESET
```

Output format for INFORMATION = *CPU

Example

```
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     DC    NO    00(RUN),01(SLEEP)
```

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 in this column, and the member CPU quota of the VM in column CPU-Q.

If the VM is not assigned to a VM group, blanks are displayed in this column, and the CPU quota of the (single) VM in column 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 VM leaves control over the real CPU 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", SU /390).

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 "Virtual CPUs" (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,FC1A,FC1B
              ASSIGNED BY GUEST
              EXCL:   CC42,CC43
              SHARED: 990F,FC1C,FC1D
```

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 within a group 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.

390 The suffix (VC) identifies a virtual console (see "Operating the guest systems (VC dialog via $VMCONS)").

Output format for INFORMATION = *MEMORY

Example (SU /390)

```
VM-ID         LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 2 SYSTEM2    00000210        512        512
```

Example (SU x86)

```
VM-ID         BS2-MEM-S  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 2 SYSTEM2        35420     36000       8192      65000
```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.

LOC-BEGIN (SU /390 only)

Start address of the main memory area in Mbytes (hexadecimal digits).

BS2-MEM-S (SU x86 only)

Size of the main memory (without the VM's firmware component) which is available for BS2000 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 (SU x86 for main memory reconfiguration). No output for SU /390.

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

Examples

```
VM-ID         VM-MONJV
 2 SYSTEM2  :4IVZ:$TSOS.JV-SYS2
VM-ID         RECONF-MONJV
 2 SYSTEM2  :4IVZ:$TSOS.JV-S2REC
VM-ID         MIGR-MONJV
 2 SYSTEM2  :4IVZ:$TSOS.JV-S2MIG
```

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, this output block is omitted.

`MIGR-MONJV`

File name of the MONJV for the VM Migration. If no VM Migration is in progress or no MONJV has been defined for this, this output block is omitted.

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

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 (SU /390)

```
/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)       512  (100) ————————  (2)
VM-ID        STATE           VERSION PER ADMIN PRIV
 2 SYSTEM2   RUNNING         V20.0B  NO  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/ 25.00 100.00/ 25.00  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,FC1A,FC1B
             ASSIGNED BY GUEST
             SHARED: 990F,FC1C,FC1D
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 2 SYSTEM2   00000210        512        512 ————————————————————————————  (7)
VM-ID        PRIV    DETAILS
 2 SYSTEM2   AG      SYS2IMPL ————————————————————————————————————————————  (8)
             AS
```

---

(1) The VM administrator queries the attributes and state of his/her VM.

(2) VM `SYSTEM2` is a monoprocessor VM (one virtual normal CPU) with a virtual spare CPU. The main memory size of the VM is 512 Mbytes. The VM belongs to the standard CPU pool.

(3) The VM is in the RUNNING state. The guest system has reached `SYSTEM READY`, the BS2000 version is output. The VM is not persistent. 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) The start address of the main memory of the VM is X'210' Mbytes, its size 512 Mbytes.

(8) The privileges of the VM.

7.5.45 SHOW-VM-CPU-POOL (Output attributes of a CPU pool)

`/SHOW-VM-CPU-POOL` is used to output the attributes of CPU pools.

The command provides 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 state
- The VMs assigned to the CPU pool and their multiprocessor level
- On SU x86: the static CPU pool `*POOL0`, see "Implementation on SU x86"

This command is reserved for the VM2000 administrator.

SHOW-**VM-CPU-POOL**

CPU-POOL-NAME = ***ALL** / ***STDPOOL** / <name 1..8> / ***BY-CPU-ID**ENTIFICATION(...) /
***BY-VM-ID**ENTIFICATION(...) / ***BY-VM-GROUP**(...)

***BY-CPU-ID**ENTIFICATION(...)

| **CPU-ID**ENTIFICATION = <x-text 1..2>

***BY-VM-ID**ENTIFICATION(...)

| **VM-ID**ENTIFICATION = <integer 1..99> / <name 1..8>

***BY-VM-GROUP**(...)

| **VM-GROUP-NAME** = <name 1..8>

,**INF**ORMATION = ***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 (SU /390).

VM-GROUP-NAME = <name 1..8>
Name of the VM group.

INFORMATION =
Specifies 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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
	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 CPU belongs to no CPU pool

Output in S variables (in user task)

The `INFORMATION` operand of the command is used to specify which S variables are to be supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
INFORMATION = *CPU	1
INFORMATION = *VM	2
INFORMATION = *SUMMARY	3
INFORMATION = *ALL	4

Output information	Name of S variable	T	Contents	Cond.
Name of the CPU pool	var(*LIST).CPU-POOL	S	*STDPOOL *POOL0 <name 1..8>	1,2,3,4
Scheduling procedure	var(*LIST).SCHEDULE	S	*NONE *BY-TIMESLICE *DEDICATED-CPUS	1,2,3,4
Number of assigned real normal CPUs	var(*LIST).NUM-OF-NORMAL-CPUS.ALL	I	<integer 0..64>	1,3,4
Number of assigned real extra CPUs	var(*LIST).NUM-OF-EXTRA-CPUS.ALL	I	<integer 0..63>	1,3,4
Real CPUs in the CPU pool	var(*LIST).REAL-CPU(*LIST).ID	S	<x-text 2..2>	1,4
State of real CPUs in the CPU pool	var(*LIST).REAL-CPU(*LIST).STA	S	*ATT / *DET / *ERR / *OFFLINE / *WAIT-FOR-MAINT	1,4
Type of real CPUs in the CPU pool	var(*LIST).REAL-CPU(*LIST).ATTR	S	*NORMAL *EXTRA *FOREIGN	1,4
Number of assigned real foreign CPUs	var(*LIST).NUM-OF-FOREIGN-CPUS.ALL	I	<integer 0..63>	1,4
Number of VMs assigned to the CPU pool	var(*LIST).NUM-OF-VMS	I	<integer 0..99>	2,3,4
Index of the VM which is assigned to the CPU pool	var(*LIST).VM(*LIST).VM-INDEX	I	<integer 1..99>	2,4
Name of the VM which is assigned to the CPU pool	var(*LIST).VM(*LIST).VM-NAME	S	<name 1..8>	2,4
Name of the VM group which is assigned to the CPU pool	var(*LIST).VM(*LIST).VM-GROUP	S	*NONE <name 1..8>	2,4
Number of attached virtual CPUs of the VM which is assigned to the CPU pool	var(*LIST).VM(*LIST).NUM-OF-ATT-CPUS	I	<integer 0..32>	2,4

Output information	Name of S variable	T	Contents	Cond.
Multiprocessor level of the VM which is assigned to the CPU pool	var(*LIST).VM(*LIST).NUM-OF-CPUS	I	<integer 1..32>	2,4
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 (SU x86)

`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 state in the format `<cpu-id>(<rstate>),...`

`<cpu-id>` CPU identifier.

`<rstate>` State of the real CPU, see "Real CPUs"
(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.

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".

7.5.46 SHOW-VM-DEFINITION (Output attributes of a VM definition)

`/SHOW-VM-DEFINITION` is used to display the attributes, resources and devices entered for a VM in the VM definition, as well as information on the automatic startup of the guest system on the VM.

The command provides the following information:

- Attributes entered for the VM in its VM definition
- Displays whether the VM definition is active, i.e. the VM is initialized
- Displays whether the high-availability (HA) attribute is reserved for the VM
- Settings for the CPU management of the VM
- Settings for the main memory of the VM
- Settings for the privileges of the VM
- Settings for automatic startup of the guest system on the VM when the VM definition is activated next time
- Devices that are assigned to the activated VM or that are to be assigned to the VM when it is activated.

This command is reserved for the VM2000 administrator. It can also be entered for initialized VMs.

SHOW-**VM-DEF**INITION
VM-NAME = ***ALL** / <name 1..8 with-wild(20)> ,**INF**ORMATION = ***STD** / ***CPU** / ***MEMORY** / ***PRIVILEGES** / ***IPL** / ***UNITS** / ***ALL**

VM-NAME =
Determines the VM definitions whose contents are to be output.

VM-NAME = *ALL
The contents of all VM definitions are output.

VM-NAME = <name 1..8 with-wild(20)>
The contents of all VM definitions whose VM name matches the specified pattern are output.

INFORMATION =
Specifies the output information.

INFORMATION = *STD
Only the entered attributes of the VM.

INFORMATION = *CPU
Only the settings for the CPU management of the VM.

INFORMATION = *MEMORY
Only the settings for the main memory of the VM.

INFORMATION = *PRIVILEGES
Only the settings for the privileges of the VM.

INFORMATION = *IPL
Only the settings for automatic startup of the guest system on the VM when the VM definition is next activated.

INFORMATION = *UNITS
Only the devices which are assigned to the VM which has already been initialized or are to be assigned to the VM when it is activated.

INFORMATION = *ALL
All the information of all outputs.

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS2312	No VM definition with appropriate name found
	64	VMS2300	VM definition not available
	64	VMS2303	VM definition cannot be accessed
	64	VMS2332	Storage for VM definitions cannot be reached (SU /390)

Output in S variables (in user task)

The `INFORMATION` operand of the command is used to specify which S variables are to be supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
INFORMATION = *STD	1
INFORMATION = *CPU	2
INFORMATION = *MEMORY	3
INFORMATION = *PRIVILEGES	4
INFORMATION = *IPL	5
INFORMATION = *UNITS	6
INFORMATION = *ALL	7

Output information	Name of S variable	T	Contents	Cond.
Name of VM	var(*LIST).VM-NAME	S	<name 1..8>	1 - 7
Index of VM (0 = *ANY)	var(*LIST).VM-INDEX	I	<integer 0..99>	1 - 7
Is the VM definition active (i.e. is the VM initialized)?	var(*LIST).ACTIVE	B	TRUE / FALSE	1 - 7

Output information	Name of S variable	T	Contents	Cond.
Does the VM have the PERSISTENT attribute?	var(*LIST).PERSISTENT	B	TRUE / FALSE	1, 7
Is it a highly available VM?	var(*LIST).HA	B	TRUE / FALSE	1, 7
Maximum I/O utilization	var(*LIST).MAX-IO-UTILIZATION	I	<integer 1..100>	1, 7
Multiprocessor level (type of specification)	var(*LIST).NUM-OF-CPUS-SPEC	S	*BY-NUMBER *MAXIMUM *EXTRA-AND-NORMAL	1, 2, 7
Multiprocessor level	var(*LIST).NUM-OF-CPUS	I	<integer 1..32>	1, 2, 7
Name of the CPU pool	var(*LIST).CPU-POOL	S	*BY-VM-GROUP *STDPOOL <name 1..8>	1, 2, 7
Name of the VM group	var(*LIST).VM-GROUP	S	*NONE <name 1..8>	1, 2, 7
Value of the CPU quota	var(*LIST).FINE-CPU-QUOTA	S	<fixed 0.01..99.99>	2, 7
Maximum CPU utilization	var(*LIST).FINE-MAX-UTIL	S	<fixed 0.01..100.00>	2, 7
Setting for VM-ACTIVE-IDLE	var(*LIST).VM-ACTIVE-IDLE	S	*NO *AT-DEDICATED-CPUS	2, 7
Size of the main memory in Mbytes	var(*LIST).MEM-SIZE	S	<integer 1..1048576>	1, 3, 7
Minimum size of the main memory in Mbytes	var(*LIST).MIN-MEM-SIZE	S	<integer 1..1048576> *UNDEF	3, 7
Maximum size of the main memory in Mbytes	var(*LIST).MAX-MEM-SIZE	S	<integer 1..1048576> *UNDEF	3, 7
Functional scope for the VM administrator	var(*LIST).ADM-PRIVIL	S	*STD / *MIN / *MAX	4, 7
Is the IO-RESET privilege assigned?	var(*LIST).IO-PRIVIL	B	TRUE / FALSE	4, 7
Is the IO-PRIORITY privilege assigned?	var(*LIST).IO-PRIO-PRIVIL	B	TRUE / FALSE	4, 7

Output information	Name of S variable	T	Contents	Cond.
Is the AUTO-SNAP-ASSIGNMENT privilege assigned?	var(*LIST).AUTO-SNAP-PRIVIL	B	TRUE / FALSE	4, 7
Is the ASSIGN-BY-GUEST privilege assigned for at least one assignment set?	var(*LIST).ASS-BY-GUEST-PRIVIL	B	TRUE / FALSE	4, 7
Specification of the assignment sets	var(*LIST).ASSIGNMENT-SET-SPEC	S	*NONE / *ALL / *EXCEPT / *LIST	4, 7
List of the assignment sets (only for *EXCEPT and *LIST)	var(*LIST).ASSIGNMENT-SET(*LIST)	S	*STDSET / <name 1..8>	4, 7
Is the guest system on the VM to be started automatically when the VM definition is activated?	var(*LIST).AUTO-IPL-PAR. EFFECTIVE	B	TRUE / FALSE	5, 7
Mnemonic name of the IPL device for the automatic start	var(*LIST).AUTO-IPL-PAR.IPL-UNIT	S	<alphanum-name 2..2> <x-text 4..4> *NONE	5, 7
Mnemonic name of the KVP output device for the automatic start	var(*LIST).AUTO-IPL-PAR.CONS-OUT	S	<alphanum-name 2..2> *NONE	5, 7
Mode for the automatic start	var(*LIST).AUTO-IPL-PAR.MODE	S	*AUTOMATIC / *FAST / *NONE	5, 7
System name for the automatic start	var(*LIST).AUTO-IPL-PAR.SYS-NAME	S	<name 1..8> *NONE	5, 7
Do the parameters of the last IPL exist?	var(*LIST).LAST-IPL	B	TRUE / FALSE	5, 7
Are the parameters of the last IPL effective?	var(*LIST).LAST-IPL-PAR.EFFECTIVE	B	TRUE / FALSE	5, 7
Mnemonic name of the IPL device at the last IPL	var(*LIST).LAST-IPL-PAR.IPL-UNIT	S	<alphanum-name 2..2> <x-text 4..4> *NONE	5, 7
Mnemonic name of the KVP output device or VC at the last IPL	var(*LIST).LAST-IPL-PAR.CONS-OUT	S	<alphanum-name 2..2> *NONE	5, 7

Output information	Name of S variable	T	Contents	Cond.
IPL mode at the last IPL	var(*LIST).LAST-IPL-PAR.MODE	S	*AUTOMATIC / *DIALOG / *FAST / *NONE	5, 7
System name at the last IPL	var(*LIST).LAST-IPL-PAR.SYS-NAME	S	<name 1..8> *NONE	5, 7
Number of device assignments	var(*LIST).NUM-OF-UNITS	I	<integer 0..32767>	6, 7
Devices of the VM	var(*LIST).UNIT(*LIST).MN	S	<alphanum-name 2..2> <x-text 4..4>	6, 7
Responsible for the device assignment [1]	var(*LIST).UNIT(*LIST).ASS-BY	S	*VM-ADMIN / *GUEST	6, 7
Assignment as pubset device?	var(*LIST).UNIT(*LIST).ASS-VIA-PUBSET	B	TRUE / FALSE	6, 7
Assignment as pubres device?	var(*LIST).UNIT(*LIST).PUBRES-FOR-ASS	B	TRUE / FALSE	6, 7
Usage mode of the device	var(*LIST).UNIT(*LIST).USAGE	S	*EXCL / *SHARED	6, 7
Is the device a virtual console?	var(*LIST).UNIT(*LIST).VC	B	TRUE / FALSE	6, 7

1 When the VM definition is not active, only the envisaged, explicit assignments by the VM2000 administrator (*VM-ADMIN) are displayed.

Output format for INFORMATION = *STD

Example

```
                          CPU-POOL(P)
VM-ID         ACT PER HA  VM-GROUP(G)   PROC   MEM-SIZE MAX-IO
 5 G4IVJ      YES NO  NO  *STDPOOL(P)    2          512    100
```

Meaning of the output columns

VM-ID

Identifies the VM by means of VM index and VM name.
If the index *ANY is entered in the VM definition of a persistent VM, a hyphen (–) is output.

ACT

Displays whether the VM definition is active, i.e. whether the VM is initialized (YES/NO).

PER

Displays whether the VM has the PERSISTENT attribute (YES/NO).

HA

Displays whether the VM has the high-availability (HA) attribute (YES/NO).

CPU-POOL(P)
VM-GROUP(G)

Displays the assignment to a VM group.
If the VM is not assigned or is not to be assigned to a VM group, the CPU pool to which the VM is assigned or is to be assigned is shown.
Whether the VM is assigned or is to be assigned to a VM group is indicated by a suffix:

<vm-group-name>(G) The VM is assigned to a VM group or is to be assigned to a VM group.
Under CPU-Q the member CPU quota of the VM is output (SU /390).

<cpu-pool-name>(P) The VM is not assigned to any VM group, or is not to be assigned to any VM group.
Under CPU-Q the CPU quota of the (individual) VM is output.

PROC

Entered virtual CPUs of the VM in the form:

m Number of virtual normal CPUs of the VM (multiprocessor level of the VM)

m (M) Number of real normal CPUs which are available on the Server Unit for VM2000 operation. The VM was or will be created with PROCESSOR=*MAXIMUM.

m (E) Number of real normal and extra CPUs which are available on the Server Unit for VM2000 operation. The VM was or will be created with PROCESSOR=*EXTRA-AND-NORMAL.

MEM-SIZE

Entered main memory size of the VM in Mbytes.

MAX-IO

Entered maximum I/O utilization of the VM.

Output format for INFORMATION = *CPU

Example

```
                  CPU-POOL(P)
VM-ID         ACT VM-GROUP(G) CPU-Q  MAX-U PROC  ACT-I
 5 G4IVJ      YES *STDPOOL(P) 70.00 100.00  2    NO
```

Meaning of the output columns

`VM-ID / ACT / CPU-POOL(P) VM-GROUP(G) / PROC`

Corresponds to `INFORMATION=*STD`.

`CPU-Q`

Entered CPU quota of the VM.

`MAX-U`

Entered maximum CPU utilization of the VM.

`ACT-I`

Entered setting for controlling the real CPU in the case of fixed CPU assignment:

`NO`	The VM leaves control over the real CPU 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", SU /390).

Output format for INFORMATION = *MEMORY

Example

```
VM-ID         ACT MEM-SIZE MIN-MEM-S MAX-MEM-S
 5 G4IVJ      YES      512       512
```

Meaning of the output columns

`VM-ID / ACT`

Corresponds to `INFORMATION=*STD`.

`MEM-SIZE`

Entered main memory size of the VM in Mbytes.

`MIN-MEM-S`

Entered minimum size of the VM’s main memory in Mbytes (for main memory reconfiguration).

`MAX-MEM-S`

Entered maximum size of the VM’s main memory in Mbytes (SU x86 for main memory reconfiguration). No output for SU /390.

Output format for INFORMATION = *PRIVILEGES

Example

```
VM-ID        ACT PRIV    DETAILS
 5 G4IVJ     YES AS
```

Meaning of the output columns

VM-ID / ACT

Corresponds to INFORMATION=*STD.

PRIV

Entered 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

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 = *IPL

Example

```
VM-ID         ACT TYPE EFF IPL-U IPL-C MODE   SYSNAME
 5 G4IVJ      YES AUTO NO
                  LAST YES FC17  CC    FAST
```

Meaning of the output columns

VM-ID / ACT

Corresponds to INFORMATION=*STD.

TYPE

Type of subsequent IPL parameters.

AUTO The IPL parameters specified using a command are displayed (AUTO-IPL parameters).

LAST If there are any, the IPL parameters are displayed with which the VM was most recently started.

EFF

Shows whether the IPL parameters are effective (YES / NO).

Notes on the meaning of the values in the output columns TYPE and EFF.

- A VM definition can contain both IPL parameters for TYPE=AUTO and for TYPE=LAST. EFF shows which entries are currently effective.
- After a VM has been initialized with /CREATE-VM, EFF=NO always applies for TYPE=AUTO.
- When a persistent VM definition is created using /CREATE-VM-DEFINITION, the values for TYPE=AUTO depend on the specifications in the AUTO-IPL=*YES/*NO parameter.
- The line with TYPE=LAST is displayed only after the first successful start of the guest system.
- The combination TYPE=AUTO and EFF=YES displays the IPL parameters for an automatic startup of the guest system on the persistent VM when its VM definition is activated next time.
- With the combination TYPE=AUTO and EFF=NO, IPL parameters for an automatic startup can already be entered in the VM definition (by a former command with AUTO-IPL=*YES). However, they are currently deactivated (by a subsequent command with AUTO-IPL=*NO). In this case, the guest system is not started automatically the next time the VM definition is activated.
- The values of the IPL parameters for TYPE=AUTO, which are entered in the VM definition, can be modified with /MODIFY-VM-ATTRIBUTES (VM definition active) or /MODIFY-VM-DEFINITION (VM definition not active) using parameter PERSISTENT=*YES(AUTO-IPL=*YES(...)/*NO) or PERSISTENT=*NO.
- With TYPE=LAST, the value for EFF cannot be controlled directly by the VM2000 administrator. In this case, the IPL parameters become effective through a successful start of the guest system (EFF=YES). They become ineffective through a normal shutdown of the guest system (EFF=NO).

IPL-U

Mnemonic device name of the disk device used for the initial program loading (IPL) of the guest system.

IPL-C

Mnemonic device name of the KVP output device or of the virtual console (only with TYPE=LAST on SU /390).

MODE

Type of system initialization (AUTO / FAST / DIALOG).

SYSNAME

System name of the guest system (SU x86). No output for SU /390.

Output format for INFORMATION = *UNITS

Example

```
VM-ID         ACT UNITS
 5 G4IVJ      YES ASSIGNED BY VM-ADMIN
                  EXCL:   CC(VC),CC48,CC49,CC88,CC89,CD48,CD49,
                          FC17(PR),FC18(PD),FC19(PD)
                  SHARED: 990F
```

Meaning of the output columns

`VM-ID / ACT`

Corresponds to `INFORMATION=*STD`.

`UNITS`

Entered 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`). Empty blocks or groups are omitted.
When no devices are to be assigned, `NO DEVICE ASSIGNED` is output.

The device list is arranged within a group 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 `(PR)` indicates the pubres device of a pubset whose devices have been assigned to the VM or are to be assigned when the VM definition is activated.

The suffix `(PD)` indicates a pubres device which has been assigned to a VM with an active VM definition (i.e. the VM is initialized) via the pubset.

> **i** When a persistent VM is deleted with `/DELETE-VM`, entries for such devices and also the entries for implicitly assigned devices (`ASSIGNED BY GUEST`) are removed from the VM definition.

390 The suffix `(VC)` identifies a virtual console (see "Operating the guest systems (VC dialog via $VMCONS)").

Example (SU /390)

```
/show-vm-definition vm-name=g4ivj,information=*all ──────────────────────  (1)
                         CPU-POOL(P)
VM-ID         ACT PER HA  VM-GROUP(G)  PROC   MEM-SIZE MAX-IO
 5 G4IVJ      YES NO  NO  *STDPOOL(P)   2          512    100
                  CPU-POOL(P)
VM-ID         ACT VM-GROUP(G) CPU-Q  MAX-U PROC  ACT-I
 5 G4IVJ      YES *STDPOOL(P) 70.00 100.00  2    NO
VM-ID         ACT MEM-SIZE MIN-MEM-S MAX-MEM-S
 5 G4IVJ      YES      512       512
VM-ID         ACT PRIV    DETAILS
 5 G4IVJ      YES AS
VM-ID         ACT TYPE EFF IPL-U IPL-C MODE   SYSNAME
 5 G4IVJ      YES AUTO NO
                  LAST YES FC17  CC    FAST
VM-ID         ACT UNITS
 5 G4IVJ      YES ASSIGNED BY VM-ADMIN
                  EXCL:   CC(VC),CC48,CC49,CC88,CC89,CD48,CD49,
                          FC17(PR),FC18(PD),FC19(PD)
                  SHARED: 990F
```

---

(1) The VM2000 administrator inquires the settings for the VM entered in the VM definition.

See also section "Examples of working with VM definitions".

7.5.47 SHOW-VM-DEVICE-STATUS (Output detailed device information)

/SHOW-VM-DEVICE-STATUS outputs detailed information on VM2000 devices.

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-FOR-ASSIGNMENT)
- 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

This command is reserved for the VM2000 administrator and VM administrators with ADMIN-PRIVILEGES=*MAXIMUM.

SHOW-**VM-DEV**ICE-**STA**TUS

UNITS = **<u>*STD</u>** / ***ALL** / ***PUB**SET-**DEV**ICES(...) /

list-poss(28): ***DEVICE-RANGE**(...) / <alphanum-name 1..4 with-wild(20)>

***PUB**SET-**DEV**ICES(...)

| **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| ***BY-PUBRES-DEV**ICE(...)

| | **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
```

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=*VM-SUMMARY`.

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>	device family (group)
*CONSOLE	Console
*PRINTER	Printer
*FAM50	Special devices
*TD	TRANSDATA device
*UR	Unit record device
*DISK	Disk 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.

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.

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 (SU /390).

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.

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 devices in the `KEEP` state (see "Assignment sets, implicit device assignment and release").

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.

ASSIGN-BY-GUEST = *NO
Outputs all devices that can only be assigned by the VM or VM2000 administrator.
Exception: Alias devices (PAV/DPAV, see "Parallel access volume (PAV, SU /390)"), CRD devices (see "Special devices") and snap units for Snapsets (see "Special devices").

ASSIGN-BY-GUEST = *YES(...)
Outputs all devices that can be assigned to a suitable VM implicitly by the guest system. Alias devices (PAV /DPAV, see "Parallel access volume (PAV, SU /390)"), CRD devices (see "Special devices") and snap units for Snapsets (see "Special devices") 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.

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.

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.

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 "Special devices").

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/DPAV (base device or alias device). See "Parallel access volume (PAV, SU /390)".

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.

> FastDPAV alias devices are displayed with selection `*ANY`, `*NO-PAV` or `*NO-ALIAS`.

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/DPAV devices (base and alias devices).

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	CMD0001	Command successfully executed; not all devices available
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

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 is used to specify which S variables are to be supplied with values. The following can be specified for INFORMATION:

Notation of command	Abbreviated form used in table
INFORMATION = *STD INFORMATION = *ALL INFORMATION = *PAV INFORMATION = *VM-SUMMARY	1 2 3 4

Output information	Name of S variable	T	Contents	Cond.
Mnemonic device name	var(*LIST).MN	S	<alphanum-name 2..2> <x-text 4..4>	1,2,3
Device type	var(*LIST).TYPE	S	<name 1..8>	1,2,3
Can the device be used as a virtual console?	var(*LIST).VC	B	TRUE / FALSE	1,2,3
Is the device configured as a snap unit?	var(*LIST).SNAP	B	TRUE / FALSE	1,2,3
Can the device be assigned implicitly by the guest system?	var(*LIST).ASS-BY-GUEST	B	TRUE / FALSE	1,2,3
Assignment set to which the device is assigned	var(*LIST).ASSIGNMENT-SET	S	<name 1..8> / *NONE / *STDSET	1,2,3
Does the device have attribute PUBRES-FOR-ATTRIBUTES?	var(*LIST).PUBRES-FOR-ATTR	B	TRUE / FALSE	1,2,3
PAV attribute of the device	var(*LIST).PAV-ATTR	S	*NO-PAV / *BASE / *ALIAS	1,2,3
Base device of the alias device (PAV/DPAV)	var(*LIST).BASE-MN	S	<alphanum-name 2..2> <x-text 4..4> *NONE	1,2,3
Alias devices of the base device (PAV/DPAV)	var(*LIST).ALIAS(*LIST).ALIAS-MN	S	<alphanum-name 2..2> <x-text 4..4>	1,2,3

Output information	Name of S variable	T	Contents	Cond.
Usage mode	var(*LIST).USAGE	S	*EXCL / *SHARED / *UNDEF	1,2,3
Type of I/O processing	var(*LIST).IO-MODE	S	*UNDEF / *DIRECT / *INDIRECT	1,2,3
Index of the VM to which the device is assigned	var(*LIST).ASS(*LIST).TO-VM-INDEX	I	<integer 1..99>	1,2,3
Name of the VM to which the device is assigned	var(*LIST).ASS(*LIST).TO-VM-NAME	S	<name 1..8>	1,2,3
Who assigned the device to the VM?	var(*LIST).ASS(*LIST).BY	S	*VM-ADMIN *GUEST	1,2,3
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
The VM index	var(*LIST).VM-INDEX	I	<integer 0..99> [1]	4
Name of VM	var(*LIST).VM-NAME	S	<name 1..8> *NONE [1]	4
Mnemonic device name	var(*LIST).UNIT(*LIST).MN	S	<alphanum-name 2..2> <x-text 4..4>	4
Who assigned the device to the VM?	var(*LIST).UNIT(*LIST).ASS-BY	S	*VM-ADMIN / *GUEST / *UNDEF	4
Usage mode	var(*LIST).UNIT(*LIST).USAGE	S	*EXCL / *SHARED / *UNDEF	4
Is the device in KEEP mode?	var(*LIST).UNIT(*LIST).KEEP	B	TRUE / FALSE	4
Is the device a virtual console?	var(*LIST).UNIT(*LIST).VC	B	TRUE / FALSE	4

[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)
CC42 HNC               SYS2IMPL     EXCL    2 SYSTEM2  GUEST
CC43 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/DPAV 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 configured as a snap unit.

`ASSIGN-SET`

Assignment set of the device.

`*STDSET / <name 1..8>`	Name of the assignment set.
`...(P)`	This suffix indicates a pubres device with attribute `PUBRES-FOR-ATTRIBUTES` (see "Assigning and removing pubsets of a VM").

`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. 390 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 "Assigning and removing pubsets of a VM").
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 "Assignment sets, implicit device assignment and release").
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.

390 For SU /390 the suffix (VC) identifies a device that is used or can be used as a virtual console (see "Operating the guest systems (VC dialog via $VMCONS)").

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)
CC42 HNC              SYS2IMPL     EXCL    2 SYSTEM2  GUEST ————————————  (4)
CC43 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)
... (output abbreviated)
5400 STDDISK  BASE
              (BASE: 5400; ALIAS: 540C)
```

---

(1) The VM2000 administrator obtains detailed information on the status of the devices of assignment set SYS2IMPL. With INFORMATION=*ALL, 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 "Assignment sets, implicit device assignment and release") 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 ("Example of peripheral device management" and "Example of pubset management").

7.5.48 SHOW-VM-GROUP (Output attributes of VM group)

x86 No VM groups are available on SU x86. 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.

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

This command is reserved for the VM2000 administrator.

SHOW-**VM-GROUP**
VM-GROUP-NAME = ***ALL** / <name 1..8> / ***BY-MEMBER**(...) / ***BY-CPU-POOL**(...) ***BY-MEMBER**(...) | **VM-ID**ENTIFICATION = <integer 1..99> / <name 1..8> ***BY-CPU-POOL**(...) | **CPU-POOL-NAME** = ***STDPOOL** / <name 1..8> ,**INF**ORMATION = ***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 codes

(SC2)	SC1	Main code	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 is used to specify which S variables are to be supplied with values. The following can be specified for `INFORMATION`:

Notation of command	Abbreviated form used in table
INFORMATION = *ATTRIBUTES	1
INFORMATION = *MEMBERS	2
INFORMATION = *ALL	3

Output information	Name of S variable	T	Contents	Cond.
Name of the VM group	var(*LIST).VM-GROUP	S	<name 1..8>	1,2,3
Name of the CPU pool	var(*LIST).CPU-POOL	S	*STDPOOL <name 1..8>	1,2,3
CPU quota of the VM group	var(*LIST).CPU-QUOTA	S	<fixed 0.01..99.99>	1,3

Output information	Name of S variable	T	Contents	Cond.
Maximum CPU utilization of the VM group	var(*LIST).MAX-CPU-UTILIZATION	S	<fixed 0.01..100.00>	1,3
Number of VMs which belong to the VM group	var(*LIST).NUM-OF-MEMBERS	I	<integer 0..99>	2,3
The VM index	var(*LIST).MEMBER(*LIST).VM-INDEX	I	<integer 1..99>	2,3
Name of VM	var(*LIST).MEMBER(*LIST).VM-NAME	S	<name 1..8>	2,3
Share of the VM in the CPU quota of the VM group	var(*LIST).MEMBER(*LIST).MEMBER-CPU-QUOTA	S	<fixed 0.01..99.99>	2,3
Effective CPU share of the VM	var(*LIST).MEMBER(*LIST).EFFECTIVE-CPU-QUOTA	S	<fixed 0.01..100.00>	2,3
Maximum CPU utilization of the VM	var(*LIST).MEMBER(*LIST).MAX-CPU-UTILIZATION	S	<fixed 0.01..100.00>	2,3
Effective maximum CPU utilization of the VM	var(*LIST).MEMBER(*LIST).EFFECTIVE-MAX-UTIL	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.

(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".

7.5.49 SHOW-VM-RESOURCES (Output status of VM2000 resources)

The status of VM2000 resources is output using `/SHOW-VM-RESOURCES`.

The command provides the following information:

- The attributes of all VMs
- The state of all initialized VMs
- The version of the BS2000 operating system in 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 (SU /390)
- 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 (SU /390)
- 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 main memory distribution over the VMs and the free main memory
- The monitoring job variables assigned to the VMs
- The overview of the Server Unit configuration
- The assignment of the Channel Subsystem Monitoring Facility (SU /390, see "Measurement with the openSM2 measurement monitor")

This command is reserved for the VM2000 administrator and VM administrators with `ADMIN-PRIVILEGES=*MAXIMUM`.

SHOW-**VM-RES**OURCES

INFORMATION = **_STD_** / ***CPU** / ***UNITS** / ***MEMORY** / ***MONJV** / ***PRIVILEGES** /
***CONF**IGURATION / ***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>

,**ATTR**IBUTES = **_*ALL_** / ***ASS**IGNED / ***FREE**

,**VM-ID**ENTIFICATION = **_*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 state of the VM.

INFORMATION = *CPU
Only information on the CPU state.

INFORMATION = *UNITS
Only the devices are specified.

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 Unit configuration.

INFORMATION = *ALL
All the information contained in the outputs *STD, *CPU, *UNITS, *MEMORY, *MONJV and *PRIVILEGES, but not in *CONFIGURATION.

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 "Output in S variables").

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> device group
*CONSOLE	Console
*PRINTER	Printer
*FAM50	Special devices
*TD	TRANSDATA device
*UR	Unit record device
*DISK	Disk 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 according to their assignment to a VM. This operand is evaluated for devices only if `UNITS=*ALL/*DEVICE-RANGE/<alphanum-name-with-wildcards>` are specified concurrently.

ATTRIBUTES = *ALL
Information for all devices or the entire main memory is output.

> Exception: The output in S variables is restricted to the main memory or those devices assigned to a VM (the same output as for `ATTRIBUTES=*ASSIGNED`)!

ATTRIBUTES = *ASSIGNED
Only the main memory or those devices assigned to a VM are output.

ATTRIBUTES = *FREE
Only the main memory or those devices not assigned to a VM (`DISPOSABLE`) are output.

VM-IDENTIFICATION =
Identification of the virtual machine.

> The operand is not evaluated for `INFORMATION=*CONFIGURATION`.

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 (SU /390).

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.

Command return codes

(SC2)	SC1	Main code	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	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

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 is used to specify which S variables are to be supplied with values. The following can be specified for INFORMATION:

Notation of command	Abbreviated form used in table
INFORMATION = *STD	1
INFORMATION = *CPU	2
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	10
INFORMATION = *MEMORY, ATTRIBUTE=*FREE	11
INFORMATION = *ALL, ATTRIBUTE=*FREE	12

Output information	Name of S variable	T	Contents	Condition
The VM index	var(*LIST).VM-INDEX	I	<integer 1..99>	1 - 5, 7 - 9
Name of VM	var(*LIST).VM-NAME	S	<name 1..8>	1 - 5, 7 - 9
Multiprocessor level of the VM	var(*LIST).NUM-OF-CPUS	I	<integer 1..32>	1,2,9
Number of virtual spare CPUs of the VM	var(*LIST).NUM-OF-VM-SPARE-CPUS	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

Output information	Name of S variable	T	Contents	Condition
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-PRIORITY 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	I	<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 VM 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 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
Is the I/O utilization monitored by IOLVM?	var(*LIST).MAX-IO-UTIL-ACTIVE	B	TRUE / FALSE	1,9

Output information	Name of S variable	T	Contents	Condition
Status of VM	var(*LIST).STA	S	*INIT-ONLY / *RUN / *WAIT / *DOWN / *SETUP	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 /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	BS2000 *UNKNOWN	1,9
Version of the guest operating system	var(*LIST).GUEST-VERSION	S	<c-string 1..8> *UNKNOWN	1,9
Does the VM have the PERSISTENT attribute?	var(*LIST).PERSISTENT	B	TRUE / FALSE	1,9
Is the VM currently being migrated?	var(*LIST).MIGRATING	B	TRUE / FALSE	1,9
Does the VM have a different I/O configuration after the migration	var(*LIST).DIFFERENT-IO-CONF	B	TRUE / FALSE	1,9
Scheduling procedure	var(*LIST).SCHEDULE	S	*NONE *BY-TIMESLICE *DEDICATED-CPUS	2,9
CPU quota (exact) of the VM group if 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

Output information	Name of S variable	T	Contents	Condition
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
Monitor job variable for monitoring the VM	var(*LIST).VM-MONJV	S	<name 1..54> / *NONE	4,9
Monitor job variable for monitoring a reconfiguration of the VM main memory	var(*LIST).RECONF-MONJV	S	<name 1..54> / *NONE	4,9
Monitor job variable for monitoring a VM migration	var(*LIST).MIGR-MONJV	S	<name 1..54> / *NONE	4,9
Command/function scope for VM2000 administrator 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	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 Unit	var(*LIST).NUM-OF-NORMAL-CPUS	I	<integer 1..64>	6
Number of real extra CPUs of the Server Unit	var(*LIST).NUM-OF-EXTRA-CPUS	I	<integer 0..63>	6
Number of real spare CPUs of the Server Unit	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

Output information	Name of S variable	T	Contents	Condition
Size of the main memory available for VM2000 in Mbytes	var(*LIST).REAL-MEM-SIZE	I	<integer 1..1048576>	6
Main memory size of domain 0 in Mbytes	var(*LIST).FOREIGN-MEM-SIZE	I	<integer 0..1048576>	6
VM2000 hypervisor main memory size in Mbytes	var(*LIST).HYPERVISOR-SIZE	I	<integer 0..255>	6
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 Unit	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	6
Status of the real CPUs of the Server Unit	var(*LIST).REAL-CPU(*LIST).STA	S	*ATT / *DET / *ERR / *OFFLINE / *WAIT-FOR-MAINT	6
Type of real CPUs of the Server Unit	var(*LIST).REAL-CPU(*LIST).ATTR	S	*NORMAL / *EXTRA / *FOREIGN / *SPARE / *UNUSED	6
Max. number of VMs for this Server Unit	var(*LIST).MAX-VMS	I	<integer 1..99>	6
Max. number of VM groups for this Server Unit	var(*LIST).MAX-VM-GROUPS	I	<integer 0..50>	6
Max. number of CPU pools for this Server Unit	var(*LIST).MAX-CPU-POOLS	I	<integer 1..64>	6
Default value for the assignment set (VM2000 parameter `ASSIGN-BY-GUEST`)	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

Output information	Name of S variable	T	Contents	Condition
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 free device in KEEP mode?	var(*LIST).UNIT(*LIST).KEEP	B	TRUE / FALSE	10,12
Largest possible main memory size for a new VM in Mbytes	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

Output format for INFORMATION = *STD

Example

```
             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  1(+1)       512  (100)
 2 SYSTEM2   *STDPOOL(P)  30.00  100.00  1(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV
 1 MONITOR   RUNNING         V21.0B  NO  YES   AG,AS
 2 SYSTEM2   RUNNING         V20.0B  NO  NO    AS
F R E E      VM: 13          LARGEST-DOMAIN(MB):   14832
```

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 "Use of IORM in VM2000 operation".

STATE

State of the VM.
If there are overlapping wait states, only the wait state with the highest priority (in the order SEL, GLB, VMA, see "Suspending a VM/resuming operation") is displayed.

INIT-ONLY	The VM is initialized (after VM initialization, after /RESET-VM or /ACTIVATE-VM-DEFINITION AUTO-IPL=*NO), but no guest system has been started yet
RUNNING	The guest system on the VM has been started and is running
WAIT(...)	The VM is in the wait state (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)

For a VM that is currently being migrated, the following states are output:

- on the source SU:
  the current state of the VM, complemented by (MIGR), e.g. RUNNING(MIGR)
- on the target SU (SU /390):
  state SETUP(MIGR)

If the I/O configuration on the target SU is different from the I/O configuration on the source SU, the current state of the migrated VM is output on the target SU, complemented by (DIFF), e.g. RUNNING(DIFF).

VERSION

Version of the BS2000 operating system in the guest system in the form Vnn.na. If the version is not (yet) known, blanks are displayed.

PER

Displays whether the VM has the PERSISTENT attribute (YES/NO).

ADMIN

Indicates if a VM administrator is connected to the virtual machine (YES/NO).

PRIV

Privileges of the VM:

```
AG    ASSIGN-BY-GUEST

AS    AUTO-SNAP-ASSIGNMENT

IO-P  IO-PRIORITY

IO-R  IO-RESET
```

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.

390 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 "Measurement with the openSM2 measurement monitor").

x86 See the information on FREE in section "Output format for INFORMATION = *MEMORY (SU x86)".

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/ 25.00 100.00/ 25.00  1(+1)
 2 SYSTEM2    *STDPOOL            30.00/ 25.00 100.00/ 25.00  1(+1)
VM-ID         SCHED ACT-I  VIRT CPUS
 1 MONITOR     DC    NO    00(RUN),01(SLEEP)
 2 SYSTEM2     DC    NO    00(RUN),01(SLEEP)
```

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 in this column, and the member CPU quota of the VM in column CPU-Q.

If the VM is not assigned to a VM group, blanks are displayed in this column, and the CPU quota of the (single) VM in column 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:

-	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 VM leaves control over the real CPU 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", SU /390).

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 "Virtual CPUs" (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 Unit, 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                       NONE
SPARE                       04(DET)
```

Meaning of the output columns

CPU-ATTR

Type of CPU (NORMAL / EXTRA / SPARE / FOREIGN / UNUSED), see "Real CPUs".

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 (SU x86).

REAL CPUS

CPU identification and state in the format <cpu-id>(<rstate>),....

<cpu-id> Identifier of the real CPU.

<rstate> State of the real CPU, see "Real CPUs"

(ATT / DET / WFM / ERR / OFF).

Output format for INFORMATION = *UNITS

Example

```
VM-ID         UNITS
 1 MONITOR    ASSIGNED BY VM-ADMIN
              EXCL:    C2,C3,C4,C5,9900,9901,9902,CC40,CC41,CC80,
                       CC81,CD40,CD41,CD80,CD81,FC10,FC11,FC12
              SHARED:  990F
 2 SYSTEM2    ASSIGNED BY VM-ADMIN
              EXCL:    CH,CI,CC42,CC43,CD42,CD43,FC1A,FC1B
DISPOSABLE    UNASS:   CA(VC),CB(VC),CC(VC),CJ,CK,CL,CM,CN,CO,CS,CT,
                       C0,C1,DH,DI,DJ,DK,DL,DM,DN,DO,DS,DT,N3,N4,N8,
                       ...
                       CD9B,CD9C,CD9D,FC13,FC14,FC15,FC16,FC17,FC18,
                       FC19,FC1C,FC1D,FC1E,FC1F
              SHARED:  990F
```

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-specific 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 which can be assigned to a VM (DISPOSABLE) is grouped in accordance with the usage mode (UNASS/SHARED/KEEP), see operand STATE=*DISPOSABLE at /SHOW-VM-DEVICE-STATUS. 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.

390 For SU /390, the suffix (VC) identifies a device that is used or can be used as a virtual console (see "Operating the guest systems (VC dialog via $VMCONS)").

If no device corresponding to the command call is assigned to the VM, 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 = *MEMORY (SU /390)

Example (VM-ID=*VM2000)

```
VM-ID         LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
HYPERVISOR    00000000         16
 1 MONITOR    00000010        512        512
 2 SYSTEM2    00000210        512        512
NONE          00000410      14832
LARGEST FREE DOMAIN:        14832
```

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 "FREE".

LOC-BEGIN

Start address of the main memory area in Mbytes.

MEM-SIZE

Size of a free memory area in Mbytes or main memory of the VM 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 (SU x86). No output for SU /390.

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 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 (SU x86)

Example (VM-ID=*VM2000)

```
VM-ID         BS2-MEM-S  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S
 1 MONITOR         8084      8192       2048      10000
 2 SYSTEM2        35420     36000       8192      65000
FOREIGN VMS                16247
FREE                       67788
```

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 domain 0.
FREE	Free main memory for the new VMs. Corresponds to the output in the last line in the case of INFORMATION=*STD, see "FREE".

> During the configuration of a VM, there is still a main memory area required for management data of the Xen hypervisor. This additional main memory is assigned automatically to a VM (siehe "Managing main memory on SU x86"). 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 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 (FOREIGN VMS) is subsequently output, followed by the free memory for the new VMs (FREE).

Output format for INFORMATION = *MONJV

Example

```
VM-ID         VM-MONJV
 2 SYSTEM2  :4IVK:$TSOS.JV-SYS2
 3 SYSTEM3  :4IVK:$TSOS.JV-SYS3
VM-ID         RECONF-MONJV
 2 SYSTEM2  :4IVK:$TSOS.JV-S2REC
VM-ID         MIGR-MONJV
 3 SYSTEM3  :4IVK:$TSOS.JV-S3MIG
```

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, this output block is omitted.

`MIGR-MONJV`

File name of the MONJV for the VM Migration.
If no VM Migration is in progress or no MONJV has been defined for this, this output block is omitted.

Output format for INFORMATION = *PRIVILEGES

Example

```
VM-ID         PRIV    DETAILS
 1 MONITOR    AG      *STDSET
              AS
 2 SYSTEM2    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

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 (SU /390)

```
MAX NUMBER OF VMS                      15
MAX NUMBER OF VM-GROUPS                 8
MAX NUMBER OF CPU-POOLS                 4
MAX NUMBER OF CPUS PER VM               4
TOTAL REAL MEMORY SIZE (MB)         15872
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                     NONE
SPARE                     04(DET)
```

Meaning of the output lines

The following configuration data of the Server Unit is displayed:

- 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.
- Default value for the implicit device assignment (*NONE/*STDSET). The default value corresponds to the VM2000 parameter ASSIGN-BY-GUEST=*NO/*YES (SU /390). On SU x86 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 (SU /390), 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 "Output format for INFORMATION = *CPU (real CPUs)".

Example (SU /390)

```
/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  1(+1)       512  (100)
 2 SYSTEM2   *STDPOOL(P)  30.00  100.00  1(+1)       512  (100)
VM-ID        STATE           VERSION PER ADMIN PRIV ──────────────────────  (3)
 1 MONITOR   RUNNING         V21.0B  NO  YES   AG,AS
 2 SYSTEM2   RUNNING         V20.0B  NO  NO    AS
F R E E      VM: 13          LARGEST-DOMAIN(MB):   14832
VM-ID        CPU-POOL  GR-CPU-Q  CPU-Q/ EFF-Q  MAX-U/ EFF-U  PROC ──────  (4)
 1 MONITOR   *STDPOOL            30.00/ 25.00 100.00/ 25.00  1(+1)
 2 SYSTEM2   *STDPOOL            30.00/ 25.00 100.00/ 25.00  1(+1)
VM-ID        SCHED ACT-I  VIRT CPUS ──────────────────────────────────────  (5)
 1 MONITOR    DC    NO    00(RUN),01(SLEEP)
 2 SYSTEM2    DC    NO    00(RUN),01(SLEEP)
CPU-ATTR     CPU-POOL     REAL CPUS ──────────────────────────────────────  (6)
NORMAL       *STDPOOL     00(ATT),01(ATT),02(ATT),03(ATT)
EXTRA                     NONE
SPARE                     04(DET)
VM-ID        VM-MONJV ──────────────────────────────────────────────────────  (7)
 2 SYSTEM2  :4IVK:$TSOS.MONJV-SYSTEM2
VM-ID        UNITS ─────────────────────────────────────────────────────────  (8)
 1 MONITOR   ASSIGNED BY VM-ADMIN
             EXCL:   C2,C3,C4,C5,9900,9901,9902,CC40,CC41,CC80,
                     CC81,CD40,CD41,CD80,CD81,FC10,FC11,FC12
             SHARED: 990F
 2 SYSTEM2   ASSIGNED BY VM-ADMIN
             EXCL:   CH,CI,CC42,CC43,CD42,CD43,FC1A,FC1B
DISPOSABLE   UNASS:  CA(VC),CB(VC),CC(VC),CJ,CK,CL,CM,CN,CO,CS,CT, ─────  (9)
                     C0,C1,DH,DI,DJ,DK,DL,DM,DN,DO,DS,DT,N3,N4,N8, ───── (10)
                     ...
                     CD9B,CD9C,CD9D,FC13,FC14,FC15,FC16,FC17,FC18,
                     FC19,FC1C,FC1D,FC1E,FC1F
             SHARED: 990F
VM-ID        LOC-BEGIN  MEM-SIZE  MIN-MEM-S  MAX-MEM-S  DEST-MEM-S ────── (11)
HYPERVISOR   00000000         16
 1 MONITOR   00000010        512        512
 2 SYSTEM2   00000210        512        512
NONE         00000410      14832
LARGEST FREE DOMAIN:       14832
VM-ID        PRIV    DETAILS ──────────────────────────────────────────── (12)
 1 MONITOR   AG      *STDSET
             AS
 2 SYSTEM2   AS
```

---

(1) The VM2000 administrator queries the state of all VM2000 resources.

(2) List of attributes of all VMs. Each VM is also assigned a virtual spare CPU.

(3) List of states of all VMs. Up to 13 further VMs can be set up. The largest free main memory size for a new VM is 14832 Mbytes.

(4) Overview of the assignment of VMs to CPU pools and VM groups, and of the shares of Server Unit 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) VM2 is monitored with a MONJV.

(8) List of devices assigned.

(9) List of free devices, sorted by usage mode. Possible virtual consoles are identified by the suffix `(VC)`.

(10) In the device lists, the 2-digit mnemonic device names come first, followed by the 4-digit ones.

(11) Distribution of the VM's main memories in VM2000 main memory.

(12) Privileges of the VMs.

7.5.50 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.

Active objects comprise attached normal and extra CPUs, active VMs, CPU pools with attached real CPUs, and VM groups with active VMs (see "Planning distribution of the CPU capacity to the VMs").

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 "Planning distribution of the CPU capacity to the VMs").

On SU /390, 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

This command is reserved for the VM2000 administrator.

SHOW-**VM-STA**TUS

INFORMATION = ***STD** / ***TIMER** / ***COUNTER** / ***SCHEDULE** / ***ALL**

,**PER**IOD = ***NO** / ***RECENT**(...) / <integer 5..3600> *seconds*

***RECENT**(...)

| **SECONDS** = <integer 1..3600> *seconds*

,**VM-ID**ENTIFICATION = ***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-ID**ENTIFICATION = ***STD** / ***NONE** / ***ALL** / list-poss(64):<x-text 1..2> / ***BY-CPU-POOL**(...)

***BY-CPU-POOL**(...)

| **CPU-POOL-NAME** = ***BY-VM-ID**ENTIFICATION / ***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 (SU /390).

INFORMATION = *SCHEDULE
The values for scheduling are output (SU /390).

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 (SU /390).

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 (SU /390).

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

390 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).

x86 No information on real CPUs is output.

CPU-IDENTIFICATION = *ALL
Information on all attached real CPUs is output (SU /390).

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 (SU /390).

CPU-IDENTIFICATION = *BY-CPU-POOL(...)
Information on the attached real CPUs in a CPU pool is output (SU /390).

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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	CMD0001	No action required (not when PERIOD=*RECENT)
2	0	VMS4403	The VM group is assigned no VM (not with PERIOD=*RECENT)
2	0	VMS4603	No CPU assigned to the CPU pool (not with PERIOD=*RECENT)
2	0	VMS4613	No VM assigned to the CPU pool (not with PERIOD=*RECENT)
	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
	64	VMS4628	CPU-ID belongs to a spare CPU

Output format for INFORMATION = *STD / *TIMER

The output consists of two information blocks:

- `CPU TIMER INFORMATION FOR REAL CPU(S)` (SU /390)
  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 (SU /390), the running VMs and their active virtual CPUs (SU /390).
  The information lines are grouped according to CPU pools, and in these according to (single) VMs and VM groups. The virtual CPUs of a VM are sorted 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 can result 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 a real CPU to or from a CPU pool
  - assigning or removing a (single) VM or a VM group to or from a CPU pool

  and on SU /390 by:
  - change of state of a virtual CPU from or to `RUNNING` state
  - 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 of the current CPU share (CUR-Q) or of the maximum CPU utilization (CUR-U).
  `ACTIVE` times then no longer need match the planned values.

When '*****' and '**#**' apply for a 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 assigned, a corresponding message (`NO CPU ACTIVE`) is output in the information line.

Example of the CPU-specific information block (SU /390)

```
CPU TIMER INFORMATION FOR REAL CPU(S)
 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
```

Meaning of the output columns

`CPU-POOL`

Name of the CPU pool.
`*STDPOOL` 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 (SU /390)

```
CPU TIMER INFORMATION FOR VM(S)
 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
```

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 Unit's current CPU capacity.
For a VM: Relative to the Server Unit'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 Unit 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 utilized as 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 Unit CPU capacity.
For a VM group: Total CPU times of all VMs in the VM group. Percentage value relative to the current Server Unit CPU capacity.
For a VM: Total CPU times of all the VM's active virtual CPUs. Percentage value relative to the current Server Unit CPU capacity.
For a virtual CPU: CPU time of the virtual CPU. Percentage value relative to the current Server Unit CPU capacity.

Example of the VM-specific information block (SU x86)

```
CPU TIMER INFORMATION FOR VM(S)
 CPU-POOL   NUM OF   CPU-Q   CUR-Q   CUR-U        VM - ACTIVE
  VM-ID       CPUS                                TIME/     %
 *POOL0          6                              47.245/ 12.12
 *STDPOOL        8          100.00  100.00       7.900/  1.52
   1 MONITOR     2   25.00   25.00   25.00       1.487/  0.28
   2 ABGAFR02    8    5.00    6.81  100.00       2.182/  0.41
   3 ABGAFR03    4   10.00   13.63   50.00       1.732/  0.33
   5 ABGAFR05    8   40.00   54.54  100.00       2.455/  0.47
```

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 Server Unit's current CPU capacity Identical to the PLAN value of 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 Unit (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 / %`

CPU time utilized as 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 Unit CPU capacity.
For a VM: CPU time of the VM (measured by Xen). Percentage value relative to the current Server Unit CPU capacity.

Output format for INFORMATION = *COUNTER (SU /390)

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 the corresponding paragraph in "Output format for INFORMATION = *STD".

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 Unit.

`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/  %    COUNT/  %
 *STDPOOL
  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 (SU /390)

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 the corresponding paragraph in "Output format for INFORMATION = *STD".

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 state 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
 *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    --
```

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 "CPU assignment in time slicing (SU /390)").

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 states 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".

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.

7.5.51 SHUTDOWN-VM (Execute shutdown in a guest system and for VM2000)

`/SHUTDOWN-VM` is used to execute a shutdown (without a restart) in the selected guest systems.

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).

The shutdown cannot be executed in a VM's guest system while the VM is being migrated (VM Migration).

> ¡ Information on command execution is provided in section "Shutdown in the guest system and of VM2000".

x86 VM2000 operation can also be terminated (analogously to coordinated shutdown) on the SU x86 via the SE Manager, see section "Terminating the monitor system and VM2000".

SHUTDOWN-VM
VM-IDENTIFICATION = ***CURR**ENT / ***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 = *CURRENT
The VM identification assigned previously by means of `/BEGIN-VM-DIALOG` is used.

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. The monitor system is the last system to be shut down. All VMs are also terminated.

The coordinated shutdown for the entire VM2000 system cannot be executed while a VM is being migrated (VM 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 state 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 "Shutdown in the guest system and of VM2000".

> The maximum wait time selected should be greater than the longest SHUTCTL wait time in the guest systems, see system parameter SHUTCTL in the “Introduction to System Administration” [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.

WAITING-TIME = <integer 1..60> *minutes*
Time in minutes during which the guest system remains in QUIET state 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, with the exception of the monitor system, and for guest systems on VMs for which a VM Migration is currently running.

VM-IDENTIFICATION = <integer 1..99> / <name 1..8>
Identifies the virtual machine by means of a VM index or VM name.

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.

390 On SU /390, the VM definitions are still processed by VM2000 when shutdown takes place. For this action BCAM must still be active in the monitor system. Consequently, when the combined shutdown of VM2000 takes place (`/SHUTDOWN-VM VM-ID=*VM2000...`), the `QUIET-TIME=0` operand should be selected.

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 state:

- 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=*NO/*END(RESTART=*NO)`. The privilege OPERATING is required for the execution of `/SHUTDOWN` (see "Privileges of the user task in the monitor system").

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed Shutdown successfully initiated
1	0	CMD0001	No action required
2	0	VMS2104	VM2000 is shut down; at least one VM is in state IN HOLD (WAIT)
	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 state IN HOLD (WAIT)
	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

Guaranteed messages

VMS2054, VMS2055, VMS2056, VMS2057, VMS2058, VMS2070, VMS2100, VMS2104, VMS3041

Example

```
/shutdown-vm vm-id=*vm2000(waiting-time=*unlimited),
                          action=*shutdown(quiet-time=0) ——————————————  (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.

See also the detailed "Example of coordinated shutdown of VM2000".

7.5.52 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, SU /390).

x86 The guest system can also be started using the SVP functions of the KVP console which is assigned to the VM.

The change of state and the specifications for IPL-UNIT, MAIN-CONSOLE and INFORMATION-BYTE are noted in the VM definition (IPL parameters of the last start).

The command cannot be issued for the monitor VM. The section "Initializing VM2000 operation" 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-CONS**OLE = ***STD** / list-poss(2): <alphanum-name 2..2>
,**HOLD** = ***NO** / ***Y**ES
,**DIAGNOSTIC-IPL** = ***NO** / ***Y**ES
,**INF**ORMATION**-BYTE** = ***FAST** / ***AUTO**MATIC / ***DIALOG**
,**PARAMS** = ***NONE** / <alphanum-name 1..8>
,**CLEAR-MEMORY** = ***NO** / ***Y**ES
,**UNL**OCK**-SAVEAREA** = ***NO** / ***Y**ES
,**VM-ID**ENTIFICATION = ***CURR**ENT / <integer 1..99> / <name 1..8>
,CHECK-VM-STATE=***Y**ES / ***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.

IPL-UNIT = *STD
The IPL device used at the last start or restart is used again. When the guest system on a VM is first started, the IPL device must be specified explicitly.

Exception for persistent VMs:

The first time the guest system on a persistent VM is started after its VM definition has been activated (without `AUTO-IPL` option), the IPL device of the VM definition (IPL parameters of the last start) is used if it is defined.

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.

390 If there is no KVP console console device pair assigned to the VM on SU /390, the virtual console of the VM is used.

Exception for persistent VMs:

The first time the guest system on a persistent VM is started after its VM definition has been activated (without `AUTO-IPL` option), the console of the VM definition (IPL parameters of the last start) is used if it is defined.

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.

390 For SU /390, 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.

x86 Only the default value may be used on SU x86.

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).

x86 On SU x86, 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.

x86 Only the default value may be used on SU x86.

The information corresponds to the specification in the PARMS field of the corresponding SVP menu screen (see section "Initializing VM2000 operation").

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.

> ¡ A guest system on a VM in SETUP(MIGR) state can neither be started with `CHECK-VM-STATE=*YES` nor with `CHECK-VM-STATE=*NO`.

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.
A VM cannot be started with `CHECK-VM-STATE=*YES` while it is being migrated (VM 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 codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
2	0	VMS2020	VM remains in state IN HOLD(WAIT)
2	0	VMS2306	VM definition could not be updated (SU /390)
	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	VMS4080	Device could not be reset
	64	VMS4084	I/O configuration modification active
	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

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.

7.5.53 SWITCH-VM-CPU (Remove CPUs from a CPU pool and add them to another CPU pool)

/SWITCH-VM-CPU is used to remove 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**

| ,**ATTR**IBUTE = ***NORMAL** / ***EXTRA**

| ,**STA**TE = ***ATT**ACHED / ***DET**ACHED / ***OFF**LINE

,**FROM-CPU-POOL** = ***STDPOOL** / <name 1..8>

,**TO-CPU-POOL** = ***STDPOOL** / <name 1..8>

,**ATTACHED-VM-CPUS** = ***ACT**IONS(...)

***ACTI**ONS(...)

| **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 SU /390, in states RUNNING, IN HOLD or SETUP(MIGR)) (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 "Number of attached real and virtual CPUs in the CPU pool". Otherwise the command is rejected. Excess virtual CPUs which have already been detached cannot then be attached again to the guest system (state BLOCK).

SOURCE-CPU-POOL = *ADJUST-NUMBER
The command is executed.

VM2000 checks the state 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 be attached again by the guest system; they are marked accordingly by VM2000 (state 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 state BLOCK for the relevant number of virtual CPUs of 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 state 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 be attached again by the guest system; they are marked accordingly by VM2000 (state of the virtual CPU in the VM2000 information commands: BLOCK).

Command return codes

(SC2)	SC1	Main code	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	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	VMS4628	Spare CPU is not assigned to any CPU pool

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, whether the number of virtual CPUs of every VM in the source CPU pool remains less than or equal to the remaining number of real CPUs. In this example, 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".

7.5.54 SWITCH-VM-DEVICES (Remove devices from one VM and add them to another)

/SWITCH-VM-DEVICES removes devices from one VM and adds 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.

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 removed and assigned devices are deleted and noted in the VM definitions of the VMs.

> The devices removed from a VM should be detached 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.

390 Alias devices (PAV/DPAV) and CRD devices cannot be removed or assigned explicitly.

This command is reserved for the VM2000 administrator.

SWITCH**-VM-DEV**ICES

UNITS = ***ALL**(...) / ***PUB**SET**-DEV**ICES(...) /

list-poss(28): ***DEVICE-RANGE**(...) / <alphanum-name 1..4 with-wild(20)>

***ALL**(...)

| **CHECK-VM-STATE** = ***YES** / ***NO**

***PUB**SET**-DEV**ICES(...)

| **PUB**SET = ***BY-PUBRES-DEV**ICE(...) / <cat-id 1..4>

| ***BY-PUBRES-DEV**ICE(...)

| | **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** / ***Y**ES

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 and CRD 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 of the specified pubset are to be removed and reassigned.

i See also "Notes on pubset specification in the VM2000 commands" and "Pubset reconfiguration".

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 are specified with *DEVICE-RANGE or with wildcards, the devices that are not assigned to the FROM-VM are ignored.

390 On SU /390, alias devices (PAV/DPAV) and CRD devices are ignored additionally.

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 that are to be removed and reassigned by the mnemonic device name.

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.
The device configuration of a VM cannot be modified with FORCE=*NO while it is being migrated (VM 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.

390 The IO-RESET privilege of the FROM-VM must be noted on SU /390 (see "Initializing a VM").

Command return codes

(SC2)	SC1	Main code	Meaning
	0	CMD0001	Command successfully executed
1	0	VMS4035	No corresponding device found
1	0	VMS4180	Specified set of devices empty
2	0	VMS2306	VM definition could not be updated (SU /390)
2	0	VMS4229	Not all devices of the pubset could be processed
	64	VMS3002	Invalid operand value (e.g. *DEVICE-RANGE) 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/DPAV)
	64	VMS4195	Device is a CRD device
	64	VMS4223	Command not executed (operand *PUBSET-DEVICES)
	64	VMS9005	Command could not be processed or could be processed only partially
	128	VMS3042	VM being migrated
	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, VMS4195`.

Example

```
/switch-vm-devices from-vm=system2,to-vm=system3,
                   units=*pubset-devices(pubset=4ivi) ————————————————  (1)
VMS4008 DEVICE 'FC1E' SWITCHED FROM VIRTUAL MACHINE (2,SYSTEM2) TO VIRTUAL
        MACHINE (3,SYSTEM3)
VMS4008 DEVICE 'FC1F' SWITCHED FROM VIRTUAL MACHINE (2,SYSTEM2) TO VIRTUAL
        MACHINE (3,SYSTEM3)
VMS4228 PUBSET '4IVI' WITH PUBRES DEVICE 'FC1E' SWITCHED FROM VM
        (2,SYSTEM2) TO VM (3,SYSTEM3) ————————————————————————————————  (2)

---------------------------------------------------------------------------------------------------------------
```

(1) The VM2000 administrator removes the pubset devices of pubset `4IVI` from VM2 and assigns them to VM3.

(2) VM2000 confirms the old and new assignments.

8 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.

8.1 Types of error in VM2000

The following types of error can occur during VM2000 operation:

- Errors in the carrier system (SU x86)
  - 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 (SU /390)
  - 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
  - BCAM failure in the monitor system

The following sections list the individual error types, outlining the corresponding message text, the cause of the error and possible corrective measures.

8.1.1 Errors in the carrier system (SU x86)

Errors in the Xen hypervisor or in the carrier system s kernel

Form	Operation of the Server Unit is terminated abnormally
Cause	Software error in the Xen hypervisor or in the carrier system’s kernel
Diagnostic materials	See the “Operation and Administration” manual [19]
Response	The Server Unit is restarted with the current settings. VM2000 is initialized again. Virtual machines 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	case a)	The component is restarted automatically. VM2000 operation continues to run without hindrance.
	case b)	The component 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 `VMS6001`. Notify Customer Support.

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	case a)	The VM2000 Agent is restarted automatically. VM2000 operation continues to run without hindrance
	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.

8.1.2 VM2000 errors

Errors during VM2000 startup

x86 On SU x86, the startup of the monitor system is continued even if an error occurs when initializing VM2000 operation.

Form	Message: `VMS50xx...,STARTUP TERMINATED`
Cause	Detected error in VM2000 startup as described in message `VMS50xx`
Diagnostic materials	CONSLOG file; SLED of the system as a whole
Response	Correct error and reload VM2000

Errors in the VM2000 hypervisor (SU /390)

x86 On SU x86, the Xen hypervisor controls all virtual machines, see "Errors in the carrier system (SU x86)".

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 reloaded automatically (see "Automatic restart following VM2000 hypervisor termination (SU /390)"). 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	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	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	CONSLOG file; automatic VMDUMP (see "VMDUMP utility routine")
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	case a)	System dump of a system task (VM2M, VM2G, VM2C, VM2J, VM2P, VM2E, VM2T)

	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)'`
		`where:`
		(&00): Name of the VM2000 module in which the error was detected
		###(&01): Flag in VM2000 module (&00)
		(&02): More detailed error information

Cause	case a)	Program error in a system task
	case b)	Internal inconsistency detected or serious VM2000 error
Diagnostic materials	case a)	CONSLOG file of the monitor system; system dump
	case b)	CONSLOG file of the monitor system; automatic VMDUMP, if created (see "VMDUMP utility routine")
Response		Continue with VM2000 operation. The affected system task is continued from a restart point (see "Restarting a task after an error in a system task of VM2000"). It may be necessary to terminate the affected VM by means of `/DELETE-VM` and to establish $VMCONS connections again.

8.1.3 Guest system errors

SETS in the BS2000 guest system

Form

Message in the BS2000 guest system:
`'crash-id: SETS; text'`

and

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`

where

`crash-id`	Crash ID of the module by which the system was terminated
`index`	VM index
`name`	VM name

Cause

Global system error in the guest system

Diagnostic materials

The SLED of the guest system can be created by means of `/START-VM ..., DIAGNOSTIC-IPL=*YES`

390 The diagnostic data of the VM2000 hypervisor is also contained in the SLED of the guest system (dump file section `VM2HYPVS`).

Response

VM2000 operation continues unaffected. The guest system can be restarted.

If an automatic restart has been defined in the guest system (see "Automatic restart after SETS in the monitor system"), 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 "Assignment sets, implicit device assignment and release" for details on how to handle implicitly assigned devices in the case of `/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 "Automatic restart after SETS in the monitor system" for details on automatic restarting of the monitor system.

390 During a failure of the monitor system on SU /390 the VM definitions are not updated (e.g. when the VM state changes, in the event of implicit device assignment). All changes are collected and executed subsequently when BCAM is active again in the monitor system (see "Working with VM definitions").

If no automatic restart is set in the monitor system, then the following applies:

390 On SU /390, 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, see "Restarting the monitor system via SVP (SU /390)".

x86 On SU x86, first the monitor system's SLED can be created. Subsequently, the monitor system can be restarted (see below and the manual "Operation and Administration" [19]).

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 (for SU /390):
	`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	390 `VMS0000`: SLED of the system as a whole `VMS0018`: SLED of the system as a whole or the monitor system x86 SLED of the monitor system

Response	390	VMS0000: Initialize VM2000 operation again VMS0018: Restart via SVP if possible, see "Restarting the monitor system via SVP (SU /390)"; otherwise, terminate all guest systems and initialize VM2000 mode again
	x86	Restarting the monitor system

Monitor system deadlock

Problems may occur in the monitor system which prevent a /SHUTDOWN command from being issued, but do not cause a SETS (monitor system is hung up or endless loop occurs, UCON BUSY, no entry possible).

390 On SU /390, the monitor system can be restarted (with SLED) via the SVP, see "Restarting the monitor system via SVP (SU /390)".

x86 On SU x86, first the monitor system's SLED can be created. Subsequently, the monitor system can be restarted (via the SE manager or the SVP functions of the assigned KVP console, see the manual "Operation and Administration" [19]).

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.

390 During a failure of the monitor system on SU /390, the VM definitions are not updated (e.g. when the VM state changes, in the event of implicit device assignment). All changes are collected and executed subsequently when BCAM is active again in the monitor system (see "Working with VM definitions").

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.

BCAM failure in the monitor system (SU /390)

After a BCAM failure in the monitor system on SU /390, VM2000 can (temporarily) no longer work with VM definitions. For the behavior of VM2000 in this case, see "Working with VM definitions".

8.2 Restart routines in VM2000

Restart routines are executed by VM2000:

- following abnormal VM2000 hypervisor termination (on SU /390),
- following abnormal VM2000 Agent termination (SU x86),
- after SETS in the monitor system,
- following abnormal task termination in one of the system tasks of VM2000.

8.2.1 Automatic restart following VM2000 hypervisor termination (SU /390)

x86 On SU x86 the Xen hypervisor controls all virtual machines, see "Errors in the carrier system (SU x86)".

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"). 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.

`/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.

8.2.2 Automatic restart following termination of the VM2000 Agent

On SU x86, the VM2000 Agent is restarted automatically after terminating abnormally. A core dump is also created in this case, see "Errors in the carrier system (SU x86)".

8.2.3 Automatic restart after SETS in the monitor system

The automatic restart system function in the monitor system makes it possible to reload the monitor system automatically after a system crash even in VM2000 operation. 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 startup of the monitor system are fulfilled, 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.

  390 If the original IPL console is replaced as main console by a redundant 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:
  - `$TSOS.SLEDFILE` (for SLED without parameter file)
  - `$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 be reloaded immediately 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. However, VM2000 and the virtual machines cannot be addressed using VM2000 commands.

390 During a failure of the monitor system on SU /390, the VM definitions are not updated (e.g. when the VM state changes, in the event of implicit device assignment). All changes are collected and executed subsequently when BCAM is active again in the monitor system (see "Working with VM definitions").

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 "Operating the guest systems (VC dialog via $VMCONS)".

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".
>
> 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.

8.2.4 Restarting the monitor system via SVP (SU /390)

x86 On SU x86, the monitor system can be restarted via the SE manager or using the SVP functions of the KVP console which is assigned to the monitor system (see manual “Operation and Administration” [19]).

Use of the restart facility in the SVP (on SU /390) requires 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, in the main menu of the SE Manager (SU /390 selected) the *SVP console* tab must be selected, the SVP console opened and in the `MODE SELECTION FRAME` screen the `RESTART` function activated.

Restart procedure

The restart of the monitor system then proceeds in the same way as an automatic restart after SETS in the monitor system (with SLED), see "Automatic restart after SETS in the monitor system".

8.2.5 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, VM2T) 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, SU /390)
  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 for monitoring a VM, main memory reconfiguration or VM migration is not executed. When the monitoring is finished, e.g. with /DELETE-VM, problems may occur in resetting the MONJV. Such a MONJV 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 restarted automatically.
- 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, SU x86)
  Requests of the VM2000 Agent may have not been executed.
- Error in the task for the VM Migration on the target SU (VM2T, SU /390) or in the dynamically created task for the VM Migration on the source SU (SU /390)
  The VM Migration is aborted.

8.3 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.

390 On SU /390, the VM2000 dump contains the data of the VM2000 monitor as well as the data of the VM2000 hypervisor.

x86 On SU x86, 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 by TSOS with `/START-VMDUMP`.

   VMDUMP can also be called by TSOS with `/START-EXECUTABLE-PROGRAM FROM-FILE=$<userid>.SYSPRG.VM2000-UTIL.<ver>`, if the user id `<userid>` under which the VM2000 dump is installed is known.
2. The input for VMDUMP is 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  INSTRUCTIONS
----------------------------
 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 SU /390).

> **i** 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 [13]).

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 SU /390, the automatic VMDUMP is also generated in the case of:

- Undefined VM2000 errors in the VM2000 hypervisor (without a message)
- Errors in the hardware (message: `VMS0017`)

On SU /390, the diagnostic data of the VM2000 hypervisor is also contained in the SLED of a guest system (dump file section `VM2HYPVS`).

8.4 Other diagnostic aids

In addition to VMDUMP, VM2000 also offers internal **trace functions**.

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 states or undefined return codes after VM2000 hypervisor calls are logged in the execution and error trace buffers.

When `*STD` is specified, important execution points in the execution trace buffer are also logged.

Specifying `*MAX` causes even more execution points in the execution trace buffer to be logged.

The following BS2000 diagnostic aids are also available for use in the monitor system:

- CONSLOG file (also contains the VM2000 commands and messages)
- REPLOG file (also contains the VM2000 REP corrections)

8.5 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].

8.5.1 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 or via the SE manager.

x86 On SU x86, the SLED of the guest system can also be created using the SVP functions of the KVP console which is assigned to the VM.

See "Assignment sets, implicit device assignment and release" for details on how to handle implicitly assigned devices in the case of `/START-VM`.

390 On SU /390, the diagnostic data of the VM2000 hypervisor is also contained in the SLEDFILE of a guest system (dump file section `VM2HYPVS`).

x86 On SU x86, 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.

390 The SLED of the monitor system can be created when the monitor system is restarted:

- Automatic restart after SETS in the monitor system
- Restarting the monitor system via SVP (SU /390)

On SU /390, the diagnostic data of the VM2000 hypervisor is also contained in the SLED of the monitor system (dump file section `VM2HYPVS`).

x86 The SLED of the monitor system can be created by:

- An automatic restart after SETS in the monitor system
- Loading the SLED explicitly via the SKP functions of the KVP console which is assigned to the monitor VM
- Using the designated action provided by the SE manager

The SLEDFILE contains no data of the Xen hypervisor or of domain 0 (X2000).

SLED for the whole VM2000 system (SU /390)

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 "Automatic restart following VM2000 hypervisor termination (SU /390)").

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.

8.5.2 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 SU /390, 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 V21.0000 SLED(21.0) from BS2000(21.0)      <  date  >  < time >


                              +00000=  001000 SCT=VM2HYPVS          W4,D  ,L19
00001000 (00000):D7C9C401  00001024  00000000  E5D4F2F0   <==>  PID▯▯▯▯▯▯▯▯▯VM20
00001010 (00010):F0F04040  E5F1F24B  F0C2F0F0  F0F00000   <==>  00  V12.0B0000▯▯
00001020 (00020):00000000  00470101  FFFFFFFF  D7C9C401   <==>  ▯▯▯▯▯▯▯▯~~~~PID▯
00001030 (00030):00000000  00000000  E2E8E2E2  E3C1D9E3   <==>  ▯▯▯▯▯▯▯▯SYSSTART
00001040 (00040):E5F2F14B  F0C2F0F0  F0F00601  00000000   <==>  V21.0B0000▯▯▯▯▯▯
00001050 (00050):00000000  00000010  E85CE3C5  E2E3C1D4   <==>  ▯▯▯▯▯▯▯▯Y*TESTAM
00001060 (00060):C5D5E35C  00000000  00800000  00E8C4D6   <==>  ENT*▯▯▯▯▯▯▯▯▯▯YDO
```

Bild 28: Output from address 1000 in the dump section VM2HYPVS

Analyzing the SLEDFILE for the whole VM2000 system (SU /390)

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 V21.0000 SLED(21.0) from VM2000(12.0)


Analyzed Object: VM2000   V12.0B0000           TID=         TSN=     W2,INF,L19
Dumpfile: :4IVF:$TSOS.SLEDFILE
                                                          (No Selection)

Generator Name: SLED     (STD)          Generator Version: <version>
Product Name:   VM2000                  Product Version:   V12.0B0000
Address of Dump-Testament: 00001024 (absolut)

Time of creating SLED:  < date >  < time >
Information about VM2000:    Hypervisor pages FROM 0000 TO 07FF
VMs created by VM2000:   HYP  VM01 VM02 VM03 VM04 VM05


VMs dumped by SLED and their page boundaries
VM01: 000800 - 0207FF




CMD:
Key: 1=Help 2=Inf 3=PCB 4=Dump   5=Dump   6=Dump   7=Dump   8=Dump   9=Dump
```

Figure 29: 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 system can be addressed with absolute addresses:

```
DAMP V21.0000 SLED(21.0) from VM2000(12.0)      <  date  >  < time >


                              +00000=  001000 ABS=00000000          W4,D  ,L19
00001000 (00000):D7C9C401  00001024  00000000  E5D4F2F0   <==>  PID▯▯▯▯▯▯▯▯▯VM20
00001010 (00010):F0F04040  E5F1F24B  F0C2F0F0  F0F00000   <==>  00  V12.0B0000▯▯
00001020 (00020):00000000  00470101  FFFFFFFF  D7C9C401   <==>  ▯▯▯▯▯▯▯▯~~~~PID▯
00001030 (00030):00000000  00000000  E2E8E2E2  E3C1D9E3   <==>  ▯▯▯▯▯▯▯▯SYSSTART
00001040 (00040):E5F2F14B  F0C2F0F0  F0F00601  00000000   <==>  V21.0B0000▯▯▯▯▯▯
00001050 (00050):00000000  00000010  E85CE3C5  E2E3C1D4   <==>  ▯▯▯▯▯▯▯▯Y*TESTAM
00001060 (00060):C5D5E35C  00000000  00800000  00E8C4D6   <==>  ENT*▯▯▯▯▯▯▯▯▯YDO
```

Figure 30: 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 V21.0000 SLED(21.0) from VM2000(12.0)      <  time  >  < date >


                              +00000=  001000 RM =00000000          W4,D  ,L19
00001000 (00000):D7C9C401  00000000  00000000  C2E2F2F0   <==>  PID▯▯▯▯▯▯▯▯▯BS20
00001010 (00010):F0F04040  E5F2F14B  F0C2F0F0  F0F00000   <==>  00  V21.0B0000▯▯
00001020 (00020):00000000  E7E5E3F4  FFFFFFFF  724BE008   <==>  ▯▯▯▯XVT4~~~~▯.▯▯
00001030 (00030):7EDC2040  710011B8  FFFFFFFF  71001274   <==>  =▯▯ ▯▯▯▯~~~~▯▯▯▯
00001040 (00040):7141B280  710011A4  7F84E220  725C39D0   <==>  ▯▯▯▯▯▯▯u"dS▯▯*▯▯
00001050 (00050):71001208  710016F0  FFFFFFFF  71001208   <==>  ▯▯▯▯▯▯▯0~~~~▯▯▯▯
00001060 (00060):FFFFFFFF  FFFFFFFF  FFFFFFFF  7FDFD000   <==>  ~~~~~~~~~~~"▯▯▯
```

Figure 31: Output from real address 1000 in a guest system

9 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.

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 a pubset:

- explicit (by the VM2000 or VM administrator)
- implicit (by the guest system)

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 envisaged for implicit assignment to a VM must have the `ASSIGN-BY-GUEST` attribute and should be assigned to precisely one assignment set.

BS2000 VM

Virtual machine with a BS2000 guest system.

Configuration Raw Device (CRD)

Special system disks for storing configuration data.

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 *VM2000* on SU /390 which controls the execution of the *guest systems* on the *virtual machines*. On SU x86, this function is executed by the Xen hypervisor.

hypervisor mode (VM2000 hypervisor mode)

Processor mode in *VM2000-HSI* on SU /390 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.

MTC

Magnetic tape cartridge

multiprocessor level

Mnemonic expression for the number of virtual CPUs of a VM.

persistent VM

VM with the PERSISTENT or “High-Availability VM” attribute. The *VM definition* of a persistent VM is still available after a reboot of the Server Unit or after `/DELETE-VM`.

real HSI

Hardware-software interface of a Server Unit which has not been supplemented by firmware components.

scheduling

When a VM is scheduled on SU /390, the VM2000 *hypervisor* starts an operable virtual CPU of a VM on a free real CPU. On SU x86, this function is executed by the Xen hypervisor.

usage mode

Type of usage of a device or a pubset:

- usable by only one VM (`EXCLUSIVE`)
- usable by multiple VMs (`SHARED`) with direct (`SH(D)`) or indirect (`SH(I)`) I/O processing

VC dialog

Synonym for the dialog of the *guest system* operator via *$VMCONS* for operating his/her *guest system*. On SU /390, the *virtual console* is required for this dialog.

virtual console

A virtual device which is generated as a physical console at hardware generation of the SU /390 but does not actually exist. If the virtual console is used as a console of the *guest system*, the message traffic is executed as a *VC dialog* via *$VMCONS*. The virtual console is not used on SU x86.

virtual CPU (virtual processor)

CPU of a *virtual machine*.

virtual HSI

Hardware-software interface of a *virtual machine*.

virtual machine, VM

A virtual Server Unit for the user on which a *guest system* runs.

virtual machine system

Program system which offers several identical or different *virtual machines* on a real Server Unit.

VM

Abbreviation for *virtual machine*.

$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).

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 definition

Configuration definition of a *VM*.

VM group

On SU /390, 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

Whole number n>=1. Identifies the *VM* as VM1 to VMn.

VM Migration

Migration of a (running) *VM* from the current Server Unit (source SU) to another Server Unit (target SU), while retaining its resources and without tangible interruption of the guest system operation. Synonym: Live Migration.

VM mode

Processor mode in *VM2000-HSI* on SU /390 in which the *virtual machines* run.

VM name

Name of a *VM* or *VM definition*. This identifies the VM in addition to the *VM index*.

VMn

Synonym for the *VM* with the *VM index* n.

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 Server Unit for an operating system (*real HSI*), on SU /390 with VM2000 extensions in the form of firmware components. The VM2000 HSI on SU /390 offers two processor modes, *(VM2000) hypervisor mode* and *VM mode*.

10 Related publications

You will find the manuals on the internet at https://bs2manuals.ts.fujitsu.com. You can order printed versions of manuals which are displayed with the order number.

[1] **BS2000 OS DX**
System Installation
User Guide

[2] **BS2000 OS DX**
Introduction to System Administration
User Guide

[3] **BS2000 OS DX**
Utility Routines
User Guide

[4] **BS2000 OS DX**
Diagnostics Handbook
User Guide

[5] **DRV**
Dual Recording by Volume
User Guide

[6] **ELSA**
Error Logging System Analysis
User Guide

[7] ***Reserved***

[8] **openNet Server**
BCAM
User Guide

[9] **openSM2**
Software Monitor
User Guide

[10] **HIPLEX MSCF**
BS2000 Systems in a Network
User Guide

[11] **SECOS**
Security Control System - Access Control
User Guide

[12] **OMNIS/OMNIS-MENU**
Functions and Commands
User Guide

[13] **BS2000 OS DX**
Commands
User Guide

[14] **IMON**
Installation Monitor
User Guide

[15] **PROP-XT** (BS2000)
Programmed Operating with SDF-P
Product Manual

[16] **BS2000 OS DX**
Executive Macros
User Guide

[17] **JV**
Job Variables
User Guide

[18] **Performance Handbook**
User Guide

[19] **Administration and Operation**
User Guide

[20] **SHC-OSD**
Storage Management for BS2000
User Guide