PowerCampus01

LPAR-Tool 1.5.0.x

What's New

Version: 1.5.0.0

Copyright © 2018-2020 by PowerCampus^{01}

This manual is the intellectual property of PowerCampus01. It may be copied as a whole or in excerpts and also printed out, as long as no parts are changed.

All informations contained in this manual have been created with great care. Nevertheless, incorrect information can not be completely ruled out. PowerCampus^{01} is not liable for any errors and their consequences. The content may be changed at any time without notice.

Software and hardware names in this manual are in many cases registered trademarks and are the copyright of the respective copyright holder.

https://www.powercampus.de

https://www.powercampus01.com

Foreword

Introduction

This handbook describes new features available in the LPAR tool starting with version 1.5.0.x. The manual presupposes the following:

- basic knowledge of working on the command line of a UNIX system
- basic understanding of virtualization concepts and features of POWER virtualization

The user manual can be downloaded from the download area on the PowerCampus^{01} website:

- *https://www.powercampus.de* or *https://www.powercampus01.com*

Additional Information

More information about the LPAR tool is available in the Tools section of the PowerCampus^{01} website:

- *https://www.powercampus.de* or *https://www.powercampus01.com*

Help with Problems

If the LPAR tool malfunctions, PowerCampus^{01} technical support can be contacted. The following URL will open a software call for the LPAR tool:

- https://www.powercampus.de/tools/lpar-tool/software-call

Support can be reached via the following e-mail address:

- E-mail: *support@powercampus.de*

Software updates of the LPAR tool can be downloaded from the download area on the PowerCampus^{01} website:

- *https://www.powercampus.de* or *https://www.powercampus01.com*

1. What’s New in 1.5.0.x

Starting with version 1.5.0.0, the command *vios* is now also implemented in C++ and supports therefore all of the features known from the other commands.

1. Integrated Test License

Version 1.5.0.0 of the LPAR-Tool, available in the download section of our website, has an integrated test license, which is valid until 30th september 2020.

The integrated test license supports up to HMCs, up to 100 managed systems and a up to 1000 LPARs.

A test license can also be requested by simply writing an email to info@powercampus.de.

2. Support for the virtual media repository

Starting with version 1.5.0.0, virtual media repositories on virtual I/O servers can be administered with the "*vios*" command. The command „*vios help rep*“ shows a list of the available keywords for the administration of virtual media repositories:

```
$ vios help rep
USAGE: vios [<option> ...] <keyword> [<option> ...] [<argument> ...]

Recognized keywords for topic 'rep' are:
   [-h <hmc>] [-m <ms>] chmedia [-v] <vios> <file_name> ro|rw|<new_name>
   [-h <hmc>] [-m <ms>] chrep [-v] <vios> <size>
   [-h <hmc>] [-m <ms>] loadopt [-f] [-r] <vios> <file_name> <vtd>
   [-h <hmc>] [-m <ms>] lsmedia [{-o <format>|-f|-j|-y}] [-F <fields>] [-s <selections>]
<vios>
   [-h <hmc>] [-m <ms>] lsrep [{-o <format>|-f|-j|-y}] [-F <fields>] [-s <selections>]
<vios>
   [-h <hmc>] [-m <ms>] lsvopt [{-o <format>|-f|-j|-y}] [-F <fields>] [-s <selections>]
<vios>
   [-h <hmc>] [-m <ms>] mkmedia [-v] <vios> <file_name> <size>
   [-h <hmc>] [-m <ms>] mkrep [-v] <vios> [<sp>] <size>
   [-h <hmc>] [-m <ms>] mkvopt [-v] <vios> <vhost> [<vtd>]
   [-h <hmc>] [-m <ms>] rmmedia [-f] [-v] <vios> <file_name>
   [-h <hmc>] [-m <ms>] rmvopt [-v] <vios> <vopt>
   [-h <hmc>] [-m <ms>] unloadopt [-r] <vios> <vtd>
$
```

Virtual media repositories can be created (*vios mkrep*), displayed (*vios lsrep*), enlarged (*vios chrep*) and deleted (*vios rmrep*). In addition, of course, virtual optical drives can be created (*vios mkvopt*), listed (*vios lsvopt*) and also deleted again (*vios rmvopt*).

The commands "*vios mkmedia*", "*vios lsmedia*", "*vios chmedia*" and "*vios rmmedia*" are used to create and manage virtual optical media.

3. Support for Storage Pools

Storage pools on virtual I/O servers can now also be administered with the "*vios*" command:

```
$ vios help sp
USAGE: vios [<option> ...] <keyword> [<option> ...] [<argument> ...]

Recognized keywords for topic 'sp' are:
   [-h <hmc>] [-m <ms>] addsppv [-f] [-v] <vios> <sp> <pv> ...
   [-h <hmc>] [-m <ms>] chbdsp [-n <new_name>] [-v] <vios> <sp> [+<size>]
   [-h <hmc>] [-m <ms>] chsp [-v] <vios> <sp> +<size>
   [-h <hmc>] [-m <ms>] lspv [{-o <format>|-f|-j|-y}] [-F <fields>] [-s <selections>]
<vios>
   [-h <hmc>] [-m <ms>] lsbdsp [{-o <format>|-f|-j|-y}] [-F <fields>] [-s <selections>]
<vios>
   [-h <hmc>] [-m <ms>] lssp [-d] [{-o <format>|-f|-j|-y}] [-F <fields>] [-s <selections>]
<vios> [<sp>]
   [-h <hmc>] [-m <ms>] lsvscsi [{-o <format>|-f|-j|-y}] [-F <fields>] [-s <selections>]
<vios> [<vhost>]
   [-h <hmc>] [-m <ms>] mkbdsp [-v] <vios> <backing_device> [<size>] [<vhost> [<vtd>]]
   [-h <hmc>] [-m <ms>] mksp [{-M|-f}] [-v] <vios> <sp> {<parent_sp> <size>|<pv> ...}
   [-h <hmc>] [-m <ms>] rmbdsp [-v] <vios> <sp> <bd>
   [-h <hmc>] [-m <ms>] rmsp [-f] [-v] <vios> <sp>
   [-h <hmc>] [-m <ms>] rmsppv [-f] [-v] <vios> <sp> <pv> ...
$
```

Both file pools and logical volume pools can be created and administered. To create a logical volume pool with the physical volumes *hdisk2* and *hdisk3* on the virtual I/O server *ms01-vio2*, the following command can be used:

```
$ vios mksp ms01-vio2 mypool1 hdisk2 hdisk3
$ vios lssp ms01-vio2 mypool1
PVNAME  PVID              SIZE
hdisk2  00fb64f2fb97a41d  952
hdisk3  00fb64f2fbc9fb32  952
$
```

Creation and simultaneous mapping of a backing device in a storage pool can be carried out with the command "*vios mkbdsp*":

```
$ vios mkbdsp ms01-vio2 mypool1 back01 100M vhost0 lpar01_hd00
$ vios lsbdsp ms01-vio2 mypool1
BDNAME  SIZE  VTD          SVSA
back01  512   lpar01_hd00  vhost0
$
```

Storage pools can be easily extended and, in the case of logical volume pools also reduced. Further details can be found in the user manual.

4. Link Aggregations

Link aggregations on virtual I/O servers can now be created and administered with the LPAR tool. The commands for the administration of link aggregations can be listed with "*vios help lnagg*":

```
$ vios help lnagg
USAGE: vios [<option> ...] <keyword> [<option> ...] [<argument> ...]

Recognized keywords for topic 'lnagg' are:
   [-h <hmc>] [-m <ms>] addlnaggadapter [-b] [-v] <vios> <lnagg> <ent>
   [-h <hmc>] [-m <ms>] chlnagg [-f] [-v] <vios> <lnagg> <attribute> ...
   [-h <hmc>] [-m <ms>] failoverlnagg [-v] <vios> <lnagg>
   [-h <hmc>] [-m <ms>] lslnagg [-a|-c] [{-o <format>|-f|-j|-y}] [-F <fields>] [-s
<selections>] <vios>
   [-h <hmc>] [-m <ms>] mklnagg [-v] <vios> <ent> ... [<attribute> ...]
   [-h <hmc>] [-m <ms>] rmlnagg [-v] <vios> <lnagg>
   [-h <hmc>] [-m <ms>] rmlnaggadapter [-v] <vios> <lnagg> <ent>
$
```

List of physical adapters that are suitable for link aggregation (candidates):

```
$ vios lslnagg -c ms05-vio1
NAME  STATUS     PHYSLOC                     PARENT  DESCRIPTION
ent1  Available  U78AA.001.ABCDE12-P1-C6-T2  pci5    4-Port 10/100/1000 Base-TX PCI-Express
Adapter (14106803)
ent2  Available  U78AA.001.ABCDE12-P1-C6-T3  pci6    4-Port 10/100/1000 Base-TX PCI-Express
Adapter (14106803)
ent5  Available  U78AA.001.ABCDE12-P1-C7-T2  pci8    4-Port 10/100/1000 Base-TX PCI-Express
Adapter (14106803)
ent6  Available  U78AA.001.ABCDE12-P1-C7-T3  pci9    4-Port 10/100/1000 Base-TX PCI-Express
Adapter (14106803)
ent3  Available  U78AA.001.ABCDE12-P1-C6-T4  pci6    4-Port 10/100/1000 Base-TX PCI-Express
Adapter (14106803)
ent7  Available  U78AA.001.ABCDE12-P1-C7-T4  pci9    4-Port 10/100/1000 Base-TX PCI-Express
Adapter (14106803)
$
```

Generate a link aggregation with the adapters *ent3* and *ent7* (Mode *IEEE802.3ad*):

```
$ vios mklnagg ms05-vio1 ent3 ent7 mode=8023ad
$
```

List all link aggregations:

```
$ vios lslnagg ms05-vio1
NAME   ADAPTERS   BACKUP  ACTIVE_CHANNEL   OPERATING_MODE                HASH_MODE
ent9   ent0,ent4  -       primary channel  Standard mode (IEEE 802.3ad)  Source and
destination TCP/UDP ports
ent13  ent3,ent7  -       primary channel  Standard mode (IEEE 802.3ad)  Source and
destination TCP/UDP ports
$
```

List all adapters in the link aggregations:

```
$ vios lslnagg -a ms05-vio1
             ADAPTER   LINK
MEDIASPEED
LNAGG  NAME  TYPE     STATUS   SELECTED          RUNNING                JUMBO_FRAMES
PHYSLOC
ent9   ent0  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C6-T1
ent9   ent4  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C7-T1
ent13  ent3  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C6-T4
ent13  ent7  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C7-T4
$
```

Remove the adapter *ent7* from the link aggregation *ent13*:

```
$ vios rmlnaggadapter ms05-vio1 ent13 ent7
$
```

List the adapters:

```
$ vios lslnagg -a ms05-vio1
             ADAPTER   LINK
MEDIASPEED
LNAGG  NAME  TYPE     STATUS   SELECTED          RUNNING                JUMBO_FRAMES
PHYSLOC
ent9   ent0  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C6-T1
ent9   ent4  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C7-T1
ent13  ent3  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C6-T4
-      ent7  primary  Up       Auto negotiation  1000 Mbps Full Duplex  Disabled
U78AA.001.ABCDE12-P1-C7-T4
$
```

Add *ent7* as a backup adapter:

```
$ vios addlnaggadapter -b ms05-vio1 ent13 ent7
$
```

5. Shared Ethernet Adapter

Shared Ethernet adapters on virtual I/O servers can now also be created and administered with the LPAR tool. Here is the list of commands:

```
$ vios help sea
USAGE: vios [<option> ...] <keyword> [<option> ...] [<argument> ...]

Recognized keywords for topic 'sea' are:
   [-h <hmc>] [-m <ms>] lssea [-a|-c] [{-o <format>|-f|-j|-y}] [-F <fields>] [-s
<selections>] <vios>
   [-h <hmc>] [-m <ms>] mksea [-v] <vios> <target_device> <ent> ... [<attribute> ...]
   [-h <hmc>] [-m <ms>] rmsea [-v] <vios> <sea>
$
```

List of physical adapters and link aggregations for a new SEA:

```
$ vios lssea -c ms05-vio1
NAME   STATUS     PHYSLOC                     PARENT  DESCRIPTION
ent1   Available  U78AA.001.ABCDE12-P1-C6-T2  pci5    4-Port 10/100/1000 Base-TX PCI-
Express Adapter (14106803)
ent2   Available  U78AA.001.ABCDE12-P1-C6-T3  pci6    4-Port 10/100/1000 Base-TX PCI-
Express Adapter (14106803)
ent5   Available  U78AA.001.ABCDE12-P1-C7-T2  pci8    4-Port 10/100/1000 Base-TX PCI-
Express Adapter (14106803)
ent6   Available  U78AA.001.ABCDE12-P1-C7-T3  pci9    4-Port 10/100/1000 Base-TX PCI-
Express Adapter (14106803)
ent13  Available  -                           -       EtherChannel / IEEE 802.3ad Link
Aggregation
$
```

List of virtual adapters for a new SEA:

```
$ vios lsdev -t vent4sea ms05-vio1
NAME   STATUS     PHYSLOC                      PARENT  DESCRIPTION
ent19  Available  U8205.E6B.XXXXXXX-V1-C7-T1   vio0    Virtual I/O Ethernet Adapter (l-lan)
ent16  Available  U8205.E6B.XXXXXXX-V1-C30-T1  vio0    Virtual I/O Ethernet Adapter (l-lan)
ent17  Available  U8205.E6B.XXXXXXX-V1-C31-T1  vio0    Virtual I/O Ethernet Adapter (l-lan)
ent14  Available  U8205.E6B.XXXXXXX-V1-C6-T1   vio0    Virtual I/O Ethernet Adapter (l-lan)
$
```

Generation of a new SEA with the link aggregation *ent13* and the virtual trunking adapters *ent14* and *ent19*:

```
$ vios mksea ms05-vio1 ent13 ent14 ent19
$
```

List all SEAs:

```
$ vios lssea ms05-vio1
                                   TIMES    TIMES   BRIDGE
NAME   HA_MODE  PRIORITY  STATE    PRIMARY  BACKUP  MODE
ent12  Auto     1         PRIMARY  1        0       All
ent15  -        -         -        -        -       -
$
```

List the adapters of all SEAs:

```
$ vios lssea -a ms05-vio1
SEA    LNAGG  NAME   TYPE     STATUS   SPEED                  VSWITCH    MODE  PHYSLOC
ent12  ent9   ent0   primary  Up       1000 Mbps Full Duplex  -          -     U78AA.001.ABCDE12-P1-
C6-T1
ent12  ent9   ent4   primary  Up       1000 Mbps Full Duplex  -          -     U78AA.001.ABCDE12-P1-
C7-T1
ent12  -      ent10  virtual  -        -                      ETHERNET0  VEB   U8205.E6B.XXXXXXX-V1-
C10-T1
ent12  -      ent11  control  -        -                      ETHCTRL    -     U8205.E6B.XXXXXXX-V1-
C11-T1
ent15  ent13  ent3   primary  UNKNOWN  Unknown                -          -     U78AA.001.ABCDE12-P1-
C6-T4
ent15  ent13  ent7   primary  UNKNOWN  Unknown                -          -     U78AA.001.ABCDE12-P1-
C7-T4
ent15  -      ent19  virtual  -        -                      ETHINNER   VEB   U8205.E6B.ABCDE12-V1-
C7-T1
ent15  -      ent14  virtual  -        -                      ETHINNER   VEB   U8205.E6B.ABCDE12-V1-
C6-T1
$
```

Distribution of the VLANs among the virtual adapters:

```
$ nvios lssea -V ms05-vio1
SEA    LNAGG  NAME   TYPE     VSWITCH    MODE  ACTIVE  PRIORITY  PVID  VLAN_TAG_IDS
ent12  ent9   ent0   primary  -          -     -       -         -     -
ent12  ent9   ent4   primary  -          -     -       -         -     -
ent12  -      ent10  virtual  ETHERNET0  VEB   True    1         1     900
ent12  -      ent11  control  ETHCTRL    -     -       -         1     None
ent15  ent13  ent3   primary  -          -     -       -         -     -
ent15  ent13  ent7   primary  -          -     -       -         -     -
ent15  -      ent19  virtual  ETHINNER   VEB   True    1         2     200,201,202
ent15  -      ent14  virtual  ETHINNER   VEB   True    1         1     100,110
$
```

6. Viewing the Error Report

The error report of a virtual I/O server can be displayed with the "*vios errlog*" command:

```
$ vios errlog ms05-vio1
IDENTIFIER  TIMESTAMP   TYPE  CLASS  RESOURCE_NAME  DESCRIPTION
CE9566DF    0611111920  P     H      ent13          TOTAL ETHERCHANNEL FAILURE
EC0BCCD4    0611111820  T     H      ent7           ETHERNET DOWN
EC0BCCD4    0611111820  T     H      ent3           ETHERNET DOWN
CE9566DF    0526145920  P     H      ent13          TOTAL ETHERCHANNEL FAILURE
39A93764    0526145920  P     H      ent7           PORT LACP FAILURE
39A93764    0526145920  P     H      ent3           PORT LACP FAILURE
EC0BCCD4    0526145720  T     H      ent7           ETHERNET DOWN
EC0BCCD4    0526145720  T     H      ent3           ETHERNET DOWN
E48A73A4    0317031820  I     H      ent12          BECOME PRIMARY
AE9D6657    0317025020  I     H      ent11          SEA NETWORK LOOP DETECTED
BE0A03E5    0316143020  P     H      sysplanar0     ENVIRONMENTAL PROBLEM
BE0A03E5    0316034420  P     H      sysplanar0     ENVIRONMENTAL PROBLEM
$
```

If you want to display more details about entries, the option "*-a*" (all details) can be used. With the option "*-n*" the number of listed entries can be limited to a desired number:

```
$ vios errlog -a -n 1 ms05-vio1
---------------------------------------------------------------------------
LABEL:            ECH_PORT_OUT_OF_SYN
IDENTIFIER:     39A93764

Date/Time:       Thu Jun 11 11:20:01 2020
Sequence Number: 128
Machine Id:      00XXXXXXXX00
Node Id:         ms05-vio1
Class:           H
Type:            PEND
WPAR:            Global
Resource Name:   ent7
Resource Class:  adapter
Resource Type:   14106803
Location:        U78AA.001.ABCDE12-P1-C7-T4

VPD:
      4-P NIC-SX PCI-e:
        EC Level....................D76641
        Part Number.................46Y3512
        Manufacture ID..............YL1026
        FRU Number..................46Y3512
        Network Address.............00145EABC123
        ROM Level.(alterable).......CL0210

Description
PORT LACP FAILURE

Probable Causes
Port did not receive LACPDU from its partner

Failure Causes
Port did not receive LACPDU from its partner

            Recommended Actions
            CHECK CABLE AND ITS CONNECTIONS
            IF ERROR PERSISTS, REPLACE ADAPTER CARD.
```

```
Detail Data
Port stopped receiving LACPDU from its partner. IEEE802.3AD port synchronization state is
OUT_OF_SYNC

$
```

7. View devices on Virtual I/O Servers

Devices on a virtual I/O server can be displayed with "*vios lsdev*":

```
$ vios lsdev ms05-vio1 ent13
NAME   STATUS     PHYSLOC  PARENT  DESCRIPTION
ent13  Available  -        -       EtherChannel / IEEE 802.3ad Link Aggregation
$
```

If no device is explicitly specified, all devices are listed:

```
$ vios lsdev ms05-vio1
NAME        STATUS     PHYSLOC      PARENT      DESCRIPTION
L2cache0    Available  -            sysplanar0  L2 Cache
VMLibrary   Available  -            rootvg      Logical volume
bd01        Available  -            rootvg      Logical volume
bd02        Available  -            rootvg      Logical volume
bd03        Available  -            rootvg      Logical volume
bd04        Available  -            rootvg      Logical volume
…
$
```

With the "*-d*" (*vital product data*) option, properties of devices that are stored on the hardware itself can be displayed:

```
$ vios lsdev -d ms05-vio1 fcs0
  fcs0             U78AA.001.ABCDE12-P1-C5-T1  8Gb PCI Express Dual Port FC Adapter
(df1000f114108a03)

        Part Number.................10N9824
        Serial Number...............XXXXXXXXXX
        Manufacturer................001C
        EC Level....................D77040
        Customer Card ID Number.....577D
        FRU Number..................10N9824
        Device Specific.(ZM)........3
        Network Address.............10000000C9010203
        ROS Level and ID............027820F5
        Device Specific.(Z0)........31004549
        Device Specific.(Z1)........00000000
        Device Specific.(Z2)........00000000
        Device Specific.(Z3)........09030909
        Device Specific.(Z4)........FF781150
        Device Specific.(Z5)........027820F5
        Device Specific.(Z6)........077320F5
        Device Specific.(Z7)........0B7C20F5
        Device Specific.(Z8)........20000000C9010203
        Device Specific.(Z9)........US2.03X5
        Device Specific.(ZA)........U2D2.03X5
        Device Specific.(ZB)........U3K2.03X5
        Device Specific.(ZC)........00000000
        Hardware Location Code......U78AA.001.ABCDE12-P1-C5-T1

  PLATFORM SPECIFIC

  Name:  fibre-channel
    Model:  10N9824
    Node:  fibre-channel@0
    Device Type:  fcp
    Physical Location: U78AA.001.WZSGB14-P1-C5-T1

$
```

Attributes of devices from the ODM can be displayed with the command "*vios lsattr*":

```
$ vios lsattr ms05-vio1 ent13
ATTRIBUTE        VALUE           DESCRIPTION                                  USER_SETTABLE
adapter_names    ent3            EtherChannel Adapters                            True
alt_addr         0x000000000000  Alternate EtherChannel Address                   True
auto_recovery    yes             Enable automatic recovery after failover         True
backup_adapter   ent7            Adapters to use when the primary channel fails   True
hash_mode        src_dst_port    Determines how outgoing adapter is chosen        True
interval         long            Determines interval value for IEEE 802.3ad mode  True
mode             8023ad          EtherChannel mode of operation                   True
netaddr          0               Address to ping                                  True
noloss_failover  yes             Enable lossless failover after ping failure      True
num_retries      3               Times to retry ping before failing               True
retry_time       1               Wait time (in seconds) between pings             True
use_alt_addr     no              Enable Alternate EtherChannel Address            True
use_jumbo_frame  no              Enable Gigabit Ethernet Jumbo Frames             True
$
```

The possible values of an attribute can also be displayed using the "*-R*" (*range*) option:

```
$ vios lsattr -R ms05-vio1 ent13 mode
standard
round_robin
8023ad
$
```

2. What’s New in 1.4.2.x

1. Integrated Test License

Version 1.4.2.0 of the LPAR-Tool, available in the download section of our website, has an integrated test license, which is valid until 30th june 2020.

The integrated test license supports up to HMCs, up to 100 managed systems and a up to 1000 LPARs.

A test license can also be requested by simply writing an email to info@powercampus.de.

2. Displaying the Model Name

The "*ms show*" command has been expanded to include the *MODEL_NAME* output column. In this column the IBM model names of the displayed managed systems are displayed:

```
$ ms show
NAME  SERIAL_NUM  TYPE_MODEL  HMCS         MODEL_NAME
ms01  XXXXXXX     8203-E4A    hmc01,hmc02  IBM Power 520 Express
ms02  XXXXXXX     9133-55A    hmc01,hmc02  IBM System p5 550Q
ms05  XXXXXXX     8233-E8B    hmc03,hmc04  IBM Power 750 Express
ms06  XXXXXXX     8406-71Y    hmc03,hmc04  IBM Power BladeCenter PS701 Express
ms11  XXXXXXX     8284-22A    hmc05,hmc06  IBM Power System S822
ms13  XXXXXXX     8286-41A    hmc05,hmc06  IBM Power System S814
ms14  XXXXXXX     8205-E6B    hmc05,hmc06  IBM Power 740 Express
ms16  XXXXXXX     9117-MMC    hmc01,hmc02  IBM Power 770
ms19  XXXXXXX     9009-22A    hmc07,hmc08  IBM Power System S922
…
$
```

3. Administration of Authorized Keys

Setting up and managing authorized keys is now very easy with the *hmc* command.

The public keys in your own *authorized_keys2* on an HMC can be listed using "*hmc lsauthkeys*":

```
$ hmc lsauthkeys hmc01
LINE
ssh-rsa AAAAB3…Jvtw== user01

$ hmc lsauthkeys hmc01 user04
LINE
ssh-rsa AAAAB3…jhvhw== user04

$
```

A public key can be added to the *authorized_keys2* on the HMC with the command "*hmc addauthkeys*":

```
$ hmc addauthkeys hmc01 "ssh-rsa AAAAkasjkjaksjf testuser"

$ hmc lsauthkeys hmc01
LINE
```

```
ssh-rsa AAAAB3…Jvtw==
ssh-rsa XXYYYZZZ
ssh-rsa AAAAkasjkjaksjf testuser

$
```

Of course, a public key can also be removed easily, using the command "*hmc rmauthkeys*":

```
$ hmc rmauthkeys hmc01 "ssh-rsa XXYYYZZZ"
$ hmc lsauthkeys hmc01
LINE
ssh-rsa AAAAB3…Jvtw==
ssh-rsa AAAAkasjkjaksjf testuser

$ hmc rmauthkeys hmc01 testuser
$ hmc lsauthkeys hmc01
LINE
ssh-rsa AAAAB3…Jvtw==

$
```

4. Initiating a System Dump

A system dump with subsequent restart can now be initiated using the "*lpar dumprestart*" command:

```
$ lpar dumprestart lpar01
$
```

The current progress of the dump can be displayed using "*lpar lsrefcode*":

```
$ lpar lsrefcode lpar01
05/08/2020 10:50:08  00cb               03400000  Dump Init:6%
05/08/2020 10:50:07  00cb               03400000  Dump Init:5%
05/08/2020 10:50:06  00cb               03400000  Dump Init:5%
05/08/2020 10:50:06  00cb               03400000  Dump Init:4%
05/08/2020 10:50:06  00cb               03400000  Dump Init:4%
05/08/2020 10:50:06  00cb               03400000  Dump Init:3%
05/08/2020 10:50:05  00cb               03400000  Dump Init:3%
05/08/2020 10:50:05  00cb               03400000  Dump Init:2%
05/08/2020 10:50:05  00cb               03400000  Dump Init:2%
05/08/2020 10:49:57  00cb               03400000  Dump Init:1%
05/08/2020 10:49:57  00cb               03400000  Dump Init:1%
05/08/2020 10:49:57  00cb               03400000  Dump Init:0%
05/08/2020 10:49:57  00cb               03400000  -
05/08/2020 10:49:57  -                  03400000  -
05/08/2020 10:49:57  D200A200           03400000  -
…
$
```

3. What’s New in 1.4.0.x

Up to version 1.2 of the LPAR tool, it was written in the C programming language. To better test the program code, we decided to rewrite the commands step-by-step in C++. In version 1.3.x the command *hmc* was already implemented in C ++. With version 1.4.x the commands *ms* and *lpar* were also implemented in C++. The functionality is retained or expanded at the same time. In the present version *1.4.0.x* the so-called unit tests have been extended to about 400. All features of the previous versions have been adopted. In addition, some useful feature and sub-commands have been added, which will be briefly described below. Detailed information can be found in the user manual.

1. Integrated Test License

To test the previous versions, it was necessary to request a so-called trial license, this was very easy by filling out a form on our web page. In version 1.3.0.0 and later, a test license is included, so that you can try the software immediately after download and installation without having to request a trial license. The integrated test license has a limited validity of 2 months from the build date of the version. On our download page it is noted how long the integrated test license is valid.

The integrated test license supports up to HMCs, up to 100 managed systems and a up to 1000 LPARs.

The test license of version 1.4.0.1 is valid until 31 october 2019.

For more extensive testing, it is still possible to request a trial license, which will support up to 4 HMCs and any number of Managed Systems and LPARs.

2. Support of different output formats

When outputting information, the output in stanza format, JSON and YAML are supported in addition to the previous standard output format. The output format will be briefly shown here using the example of the command "*hmc lshmcusr*". First the standard output format:

```
$ hmc lshmcusr hmc01
NAME     DESCRIPTION       TASKROLE       RESOURCEROLE
root     root              hmcsuperadmin  ALL:
as       Armin Schmidt     hmcsuperadmin  ALL:
hscroot  HMC Super User    hmcsuperadmin  ALL:
$
```

For output in stanza format, the option "*-f*" or "*-o stanza*" can be used:

```
$ hmc lshmcusr -f hmc01
root:
     authentication_type = local
     description = root
     disabled = 0
     idle_timeout = 0
     inactivity_expiration = 0
     min_pwage = 0
     name = root
     password_encryption = sha512
     pwage = 99999
     remote_ssh_access = 1
     remote_webui_access = 1
     resourcerole = ALL:
     resources = <ResourceID = ALL:><UserDefinedName = AllSystemResources>
     session_timeout = 0
     taskrole = hmcsuperadmin
     verify_timeout = 15
as:
     authentication_type = local
     description = Armin Schmidt
     disabled = 0
     idle_timeout = 0
     inactivity_expiration = 0
     min_pwage = 0
     name = as
     password_encryption = sha512
     pwage = 99999
     remote_ssh_access = 1
     remote_webui_access = 1
     resourcerole = ALL:
     resources = <ResourceID = ALL:><UserDefinedName = AllSystemResources>
     session_timeout = 0
     taskrole = hmcsuperadmin
     verify_timeout = 15
hscroot:
     authentication_type = local
     description = HMC Super User
     disabled = 0
     idle_timeout = 0
     inactivity_expiration = 0
     min_pwage = 0
     name = hscroot
     password_encryption = sha512
     pwage = 99999
     remote_ssh_access = 1
     remote_webui_access = 1
     resourcerole = ALL:
     resources = <ResourceID = ALL:><UserDefinedName = AllSystemResources>
     session_timeout = 0
     taskrole = hmcsuperadmin
     verify_timeout = 15
$
```

In contrast to the standard output format, all available attributes are displayed here. The attribute names are the names used by default on the HMC.

An output in JSON format can be obtained with the option "*-j*" or "*-o json*":

```
$ hmc lshmcusr -o json hmc01
{
     "authentication_type": "local",
     "description": "root",
     "disabled": "0",
     "idle_timeout": "0",
     "inactivity_expiration": "0",
     "min_pwage": "0",
     "name": "root",
     "password_encryption": "sha512",
     "pwage": "99999",
     "remote_ssh_access": "1",
     "remote_webui_access": "1",
     "resourcerole": "ALL:",
     "resources": "<ResourceID = ALL:><UserDefinedName = AllSystemResources>",
     "session_timeout": "0",
     "taskrole": "hmcsuperadmin",
     "verify_timeout": "15"
}
{
     "authentication_type": "local",
     "description": "Armin Schmidt",
     "disabled": "0",
     "idle_timeout": "0",
     "inactivity_expiration": "0",
     "min_pwage": "0",
     "name": "as",
     "password_encryption": "sha512",
     "pwage": "99999",
     "remote_ssh_access": "1",
     "remote_webui_access": "1",
     "resourcerole": "ALL:",
     "resources": "<ResourceID = ALL:><UserDefinedName = AllSystemResources>",
     "session_timeout": "0",
     "taskrole": "hmcsuperadmin",
     "verify_timeout": "15"
}
{
     "authentication_type": "local",
     "description": "HMC Super User",
     "disabled": "0",
     "idle_timeout": "0",
     "inactivity_expiration": "0",
     "min_pwage": "0",
     "name": "hscroot",
     "password_encryption": "sha512",
     "pwage": "99999",
     "remote_ssh_access": "1",
     "remote_webui_access": "1",
     "resourcerole": "ALL:",
     "resources": "<ResourceID = ALL:><UserDefinedName = AllSystemResources>",
     "session_timeout": "0",
     "taskrole": "hmcsuperadmin",
     "verify_timeout": "15"
}
$
```

If required, output can also be in YAML format using the option "*-y*" or "*-o yaml*":

```
$ hmc lshmcusr -y hmc01
---
  authentication_type: local
  description: root
  disabled: 0
  idle_timeout: 0
  inactivity_expiration: 0
  min_pwage: 0
  name: root
  password_encryption: sha512
  pwage: 99999
  remote_ssh_access: 1
  remote_webui_access: 1
  resourcerole: ALL:
  resources: <ResourceID = ALL:><UserDefinedName = AllSystemResources>
  session_timeout: 0
  taskrole: hmcsuperadmin
  verify_timeout: 15
```

```
---
  authentication_type: local
  description: Armin Schmidt
  disabled: 0
  idle_timeout: 0
  inactivity_expiration: 0
  min_pwage: 0
  name: as
  password_encryption: sha512
  pwage: 99999
  remote_ssh_access: 1
  remote_webui_access: 1
  resourcerole: ALL:
  resources: <ResourceID = ALL:><UserDefinedName = AllSystemResources>
  session_timeout: 0
  taskrole: hmcsuperadmin
  verify_timeout: 15
---
  authentication_type: local
  description: HMC Super User
  disabled: 0
  idle_timeout: 0
  inactivity_expiration: 0
  min_pwage: 0
  name: hscroot
  password_encryption: sha512
  pwage: 99999
  remote_ssh_access: 1
  remote_webui_access: 1
  resourcerole: ALL:
  resources: <ResourceID = ALL:><UserDefinedName = AllSystemResources>
  session_timeout: 0
  taskrole: hmcsuperadmin
  verify_timeout: 15
$
```

Currently, these output formats are available for the *hmc*, *ms* and *lpar* command. A corresponding extension of the *vios* command will be available shortly.

3. Selection of the Output Format

Which data records are to be output for output commands can be determined by means of selections. You can use string comparisons, regular expressions or numeric comparisons.

Comparison with a string:

```
$ lpar status -s state=Running
NAME                  LPAR_ID  LPAR_ENV   STATE    PROFILE   SYNC  RMC       PROCS
PROC_UNITS  MEM    OS_VERSION
aix01                 9        aixlinux   Running  standard  0     active    2
0.2         24576  AIX 7.1 7100-05-02-1810
aix03                 22       aixlinux   Running  standard  0     active    2
0.2         24576  AIX 7.1 7100-05-02-1810
…
$
```

Or LPARs not in the *Running* state:

```
$ lpar status -s state!=Running
NAME                  LPAR_ID  LPAR_ENV  STATE          PROFILE   SYNC  RMC       PROCS
PROC_UNITS  MEM   OS_VERSION
aix02                 18       aixlinux  Not Activated  -         0     inactive  0      -
           0     Unknown
aix04                 26       aixlinux  Not Activated  standard  0     inactive  1
0.4         1024  Unknown
…
$
```

Regular expressions:

```
$ lpar status -s os_version~7100-04
NAME           LPAR_ID  LPAR_ENV  STATE    PROFILE   SYNC  RMC     PROCS  PROC_UNITS
MEM    OS_VERSION
lpar11         19       aixlinux  Running  standard  0     active  3      0.9
81920  AIX 7.1 7100-04-04-1717
lpar13         19       aixlinux  Running  standard  0     active  3      0.9
81920  AIX 7.1 7100-04-04-1717
…
$
```

Numerical comparisons are also possible:

```
$ lpar lsmem -s curr_mem:gt:32768
              MEMORY          MEMORY           HUGE_PAGES
LPAR_NAME    MODE  AME  MIN   CURR   MAX     MIN  CURR  MAX
lpar11       ded   1.0  1024  81920  163840  0    0     0
lpar13       ded   1.0  1024  81920  163840  0    0     0
…
$
```

Besides *gt* (greater than), *ge* (greater or equal), *eq* (equal), *ne* (not equal), *lt* (less than) and *le* (less or equal) are supported. The chosen comparison is placed between 2 colons.

Of course, this can be combined with all supported output formats.

4. Selection of the Output Fields

Which data fields (or attributes) are to be shown for output commands can be determined with the help of the new *-F* option. This makes it easy to implement your own output formats.

```
$ lpar lsmem -F lpar_name:curr_mem
aix11:0
lpar9:1024
lpar11:81920
…
$
```

This can of course be combined with selections and all supported output formats:

```
$ lpar lsmem -s curr_mem:lt:4096 -y -F lpar_name:curr_mem
---
  curr_mem: 0
  lpar_name: aix11
---
```

```
  curr_mem: 1024
  lpar_name: lpar9
---
  curr_mem: 0
  lpar_name: aix02
…
$
```

The selection of data fields can be combined with all output formats and the selection of data sets.

5. Support for SR-IOV

The administration of SR-IOV is supported by the following new sub-commands:

- *ms chsriov* - Change the attributes of SR-IOV adapters and physical ports
- *ms lssriov* - List attributes of SR-IOV adapters, physical and logical ports
- *lpar addsriov* - Add an SR-IOV logical port to an LPAR
- *lpar chsriov* - Change the attributes of an SR-IOV logical port
- *lpar lssriov* - Display attributes of SR-IOV logical ports of an LPAR
- *lpar rmsriov* - Remove one SR-IOV logical port from an LPAR R

6. Support for vNICs

The administration of vNICs is supported by the following new sub-commands:

- *lpar addvnic* - Add a vNIC adapter to an LPAR
- *lpar addvnicbkdev* - Add a vNIC backing device
- *lpar rmvnicbkdev* - Removing a vNIC BackingDevices
- *lpar rmvnic* - removing a vNIC adapter

7. Administration of VLAN-IDs

The administration of VLAN IDs of virtual Ethernet adapters has been simplified. VLAN IDs can now be added or removed with the following sub-commands:

- *lpar addvlan* - Add VLANs to a virtual adapter
- *lpar rmvlan* - Removing VLANs from a virtual adapter

8. Performance-Optimization for some Sub-Commands

For many sub-commands that display information for multiple managed systems or LPARs, performance has increased by a factor of 10 in some cases.