



PowerCampus⁰¹
LPAR-Tool 1.4.2.x
Neuerungen

Version: 1.4.2.0

Copyright © 2018-2020 by PowerCampus⁰¹

Dieses Handbuch ist geistiges Eigentum von PowerCampus⁰¹.

Es darf als Ganzes oder in Auszügen kopiert und auch ausgedruckt werden, solange keine Teile verändert werden.

Alle in diesem Handbuch enthaltenen Informationen wurden mit größter Sorgfalt zusammengestellt. Dennoch können fehlerhafte Angaben nicht gänzlich ausgeschlossen werden. PowerCampus⁰¹ haftet nicht für eventuelle Fehler und deren Folgen. Der Inhalt kann jederzeit ohne Ankündigung geändert werden.

Soft- und Hardwarebezeichnungen in diesem Handbuch sind in vielen Fällen eingetragene Warenzeichen und unterliegen dem Copyright der jeweiligen Hersteller.

<https://www.powercampus.de>

<https://www.powercampus01.com>

Vorwort	4
Einführung.....	4
Weitere Informationen.....	4
Hilfe bei Problemen.....	4
1. Neuerungen in 1.4.2.x	5
1. Integrierte Test-Lizenz.....	5
2. Anzeige des Model-Namens	5
3. Administration von Authorized Keys.....	5
4. Durchführung eines System-Dumps	6
2. Neuerungen in 1.4.0.x	7
1. Integrierte Test-Lizenz.....	7
2. Unterstützung verschiedener Ausgabeformate	7
3. Auswahl der auszugebenden Datensätze	10
4. Auswahl der auszugebenden Datenfelder	11
5. Unterstützung von SR-IOV.....	12
6. Unterstützung von vNICs	12
7. VLAN-IDs Administrieren	12
8. Performance-Verbesserungen bei einigen Sub-Kommandos	13

Vorwort

Einführung

Dieses Handbuch beschreibt die Neuerungen der Version 1.4.2.x des LPAR-Tools. Das Handbuch setzt folgendes voraus:

- grundlegende Kenntnis zum Arbeiten auf der Kommandozeile eines UNIX Systems
- grundlegendes Verständnis der Virtualisierungskonzepte und Funktionen der POWER Virtualisierung

Das Benutzerhandbuch kann über den Download-Bereich auf der PowerCampus⁰¹ Webseite heruntergeladen werden:

- <https://www.powercampus.de> oder <https://www.powercampus01.com>

Weitere Informationen

Weitere Informationen zum LPAR-Tool sind im Tool-Bereich auf der PowerCampus⁰¹ Webseite verfügbar:

- <https://www.powercampus.de> oder <https://www.powercampus01.com>

Hilfe bei Problemen

Bei fehlerhaftem Funktionieren des LPAR-Tools kann der technische Support von PowerCampus⁰¹ kontaktiert werden. Über die folgende URL kann ein Software-Call für das LPAR-Tool eröffnet werden:

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

Der Support ist über die nachfolgende Email-Adresse erreichbar:

- E-mail: support@powercampus.de

Software Updates des LPAR-Tools können über den Download-Bereich auf der PowerCampus⁰¹ Webseite heruntergeladen werden:

- <https://www.powercampus.de> oder <https://www.powercampus01.com>

1. Neuerungen in 1.4.2.x

1. Integrierte Test-Lizenz

Version 1.4.2.0 des LPAR-Tools, verfügbar im Download-Bereich unserer Website, hat eine integrierte Test-Lizenz, welche bis 30.06.2020 gültig ist.

Die integrierte Test-Lizenz unterstützt bis zu 10 HMCs, bis zu 100 Managed Systems und bis zu 1000 LPARs.

Benötigen Sie eine Test-Lizenz über den 30.06.2020 hinaus, können Sie eine solche jederzeit erhalten. Schreiben Sie einfach eine Email an info@powercampus.de.

2. Anzeige des Model-Namens

Das Kommando „*ms show*“ wurde um die Ausgabe-Spalte *MODEL_NAME* erweitert. In dieser Spalte wird der IBM Modell-Name der angezeigten Managed Systems ausgegeben:

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

3. Administration von Authorized Keys

Das Einrichten und Verwalten von Authorized Keys ist nun ganz einfach mit dem *hmc*-Kommando möglich.

Die Public Keys in der eigenen *authorized_keys2* auf einer HMC können mittels „*hmc lsauthkeys*“ aufgelistet werden:

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

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

$
```

Ein Public Key kann mit dem Kommando „*hmc addauthkeys*“ einer *authorized_keys2* auf der HMC hinzugefügt werden:

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

$ hmc lsauthkeys hmc01
LINE
ssh-rsa AAAAB3...Jvtw==
ssh-rsa XXYYYYZZZ
ssh-rsa AAAAkasjkjaksjf testuser

$
```

Natürlich kann ein Public Key genauso einfach auch wieder entfernt werden, Kommando „*hmc rmauthkeys*“:

```
$ hmc rmauthkeys hmc01 "ssh-rsa XXYYYYZZZ"
$ 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. Durchführung eines System-Dumps

Ein System-Dump mit anschließendem Restart kann nun über das Kommando „*lpar dumprestart*“ initiiert werden:

```
$ lpar dumprestart lpar01
$
```

Der aktuelle Fortschritt des Dumps kann mittels „*lpar lsrefcode*“ angezeigt werden:

```
$ 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 -
...
$
```

2. Neuerungen in 1.4.0.x

Bis zur Version 1.2 des LPAR-Tools war dieses in der Programmiersprache C geschrieben. Um den Programm-Code noch besser testen zu können, haben wir uns entschieden die Kommandos schrittweise in C++ umzuschreiben. In der Version 1.3.x war schon das Kommando *hmc* neu in C++ implementiert. Mit der Version 1.4.x wurden auch die Kommandos *ms* und *lpar* neu in C++ implementiert. Die Funktionalität bleibt dabei erhalten bzw. wird gleichzeitig erweitert. In der vorliegenden Version 1.4.0.x wurden die sogenannten Unit-Tests auf ca 400 erweitert. Alle Feature der bisherigen Versionen sind dabei übernommen worden. Zusätzlich sind einige nützliche Feature und Sub-Kommandos hinzugefügt worden, welche im weiteren kurz beschrieben werden. Ausführliche Informationen finden sich im Benutzerhandbuch.

1. Integrierte Test-Lizenz

Zum Testen der bisherigen Versionen war es notwendig eine sogenannte Trial-Lizenz anzufordern, dies war durch Ausfüllen eines Formblatts auf unserer Web-Seite sehr einfach möglich. In der Version 1.3.0.0 und später ist eine Test-Lizenz enthalten, so dass man die Software unmittelbar nach dem Download und der Installation ausprobieren kann, ohne eine Lizenz anfordern zu müssen. Die integrierte Test-Lizenz hat eine beschränkte Gültigkeit von maximal 2 Monaten ab dem Build-Datum der Version. Auf unserer Download-Seite ist jeweils vermerkt bis wann die integrierte Test-Lizenz gültig ist.

Die integrierte Test-Lizenz unterstützt bis zu 10 HMCs, bis zu 100 Managed Systems und bis zu 1000 LPARs.

Die Test-Lizenz der Version 1.4.0.1 ist bis zum 31 Oktober 2019 gültig.

2. Unterstützung verschiedener Ausgabeformate

Bei der Ausgabe von Informationen werden zusätzlich zum bisherigen Standard-Ausgabeformat die Ausgabe in Stanza-Format, JSON und YAML unterstützt. Die Ausgabeformate sollen hier kurz am Beispiel des Kommandos „*hmc lshmcusr*“ gezeigt werden. Zunächst das Standard-Ausgabeformat:

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

Für die Ausgabe in Stanza-Format kann die Option „-f“ oder „-o stanza“ verwendet werden:

```
$ 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
$

```

Im Gegensatz zum Standard-Ausgabeformat werden hier alle verfügbaren Attribute angezeigt. Die Attribut-Namen sind die standardmäßig auf der HMC verwendeten Attribut-Namen.

Eine Ausgabe im JSON-Format erhält man mit der Option „-j“ oder „-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"
}
$

```

Bei Bedarf kann die Ausgabe auch im YAML-Format erfolgen, hierzu muss die Option „-y“ oder „-o yam!“ verwendet werden:

```

$ 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
$

```

Aktuell stehen diese Ausgabeformate für die Kommandos *hmc*, *ms* und *lpar* zur Verfügung. Eine entsprechende Erweiterung des anderen Kommandos *vios* wird in Kürze verfügbar sein.

3. Auswahl der auszugebenden Datensätze

Welche Datensätze bei Ausgabekommandos ausgegeben werden sollen, kann mit Hilfe von Selektionen bestimmt werden. Dabei können Vergleiche von Zeichenketten, reguläre Ausdrücke oder numerische Vergleiche verwendet werden.

Vergleich mit einer Zeichenkette:

```

$ 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
...
$

```

oder LPARs im Zustand nicht-*Running*:

```
$ 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
...
$
```

Reguläre Ausdrücke:

```
$ 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
...
$
```

Auch numerische Vergleiche sind möglich:

```
$ 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
...
$
```

Neben *gt* (greater than), sind auch *ge* (greater or equal), *eq* (equal), *ne* (not equal), *lt* (less than) und *le* (less or equal) möglich. Der gewählte Vergleich steht dabei jeweils zwischen 2 Doppelpunkten.

Natürlich lässt sich dies mit allen unterstützten Ausgabeformaten kombinieren.

4. Auswahl der auszugebenden Datenfelder

Welche Datenfelder (bzw. Attribute) bei Ausgabekommandos ausgegeben werden sollen, kann mit Hilfe der neuen Option *-F* bestimmt werden. Damit können eigene Ausgabeformate leicht realisiert werden.

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

Dies kann natürlich auch mit Selektionen und einem beliebigen Ausgabeformat kombiniert werden:

```
$ 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
...
$
```

Die Auswahl der Datenfelder lässt sich mit allen Ausgabeformaten und der Selektion von Datensätzen beliebig kombinieren.

5. Unterstützung von SR-IOV

Die Administration von SR-IOV wird über die folgenden neuen Sub-Kommandos unterstützt:

- *ms chsriov* - Ändern der Attribute von SR-IOV Adaptern und physikalischen Ports
- *ms lssriov* - Auflisten von Attributes von SR-IOV Adaptern, physikalischen und logical Ports
- *lpar addsriov* - Hinzufügen eines SR-IOV logical Ports zu einer LPAR
- *lpar chsriov* - Ändern der Attribute eines SR-IOV logical Ports
- *lpar lssriov* - Anzeigen von Attributen von SR-IOV logical Ports einer LPAR
- *lpar rmsriov* - Entfernen eines SR-IOV logical Ports aus einer LPAR

6. Unterstützung von vNICs

Die Administration von vNICs wird über die folgenden neuen Sub-Kommandos unterstützt:

- *lpar addvnic* - Hinzufügen eines vNIC Adapters zu einer LPAR
- *lpar addvnicbkdev* - Hinzufügen eines vNIC Backing Devices
- *lpar rmvnicbkdev* - Wegnehmen eines vNIC BackingDevices
- *lpar rmvnic* - Wegnehmen eines vNIC Adapters

7. VLAN-IDs Administrieren

Die Administration von VLAN-IDs von virtuellen Ethernet Adaptern wurde vereinfacht. VLAN-IDs können nun mit den folgenden Sub-Kommandos hinzugefügt oder weggenommen werden:

- *lpar addvlan* - Hinzufügen von VLANs zu einem virtuellen Adapter
- *lpar rmvlan* - Wegnehmen von VLANs von einem virtuellen Adapter

8. Performance-Verbesserungen bei einigen Sub-Kommandos

Bei vielen Sub-Kommandos, die Informationen für mehrere Managed Systems oder LPARs anzeigen, konnte die Performance teilweise um das 10 fache gesteigert werden.