PowerCampus[01]

LPAR-Tool 1.2.x

User Guide

Version: 1.2.0

https://www.powercampus.de

Foreword

Introduction

This user guide is intended for administrators and users who use the LPAR tool to administer and configure POWER virtualization. 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 PowerCampus01 website:

- *https://www.powercampus.de*

Additional Information

More information about the LPAR tool is available in the Tools section of the PowerCampus01 website:

- *https://www.powercampus.de*

Help with Problems

If the LPAR tool malfunctions, PowerCampus01 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 PowerCampus01 website:

- *https://www.powercampus.de*

1. Introduction

The LPAR tool essentially consists of the 4 programs *hmc*, *ms*, *lpar* and *vios*. These programs can be used from the command line to administer a POWER virtualization environment. Most tasks that are otherwise performed via the HMC GUI can be performed conveniently and efficiently from the command line with these commands.

1. Prerequisites

The LPAR tool is available in versions for AIX, Linux and MacOS (in preparation).

Necessary prerequisite for the functionality of the LPAR tool is an SSH connection to the HMC(s). By default, the LPAR tool uses the account of the logged-in user as the HMC user. If the HMC user is another user, this can be specified.

The SSH login to the HMC should be possible without entering a password or passphrase, which can be achieved by using the *ssh-agent*. The user's public key must be previously stored on the HMC. If no *ssh-agent* is used, the LPAR tool can still be used, but a password or passphrase prompt will be requested each time the tool is called.

A valid license is required to use the LPAR tool.

2. Installation

There are two different ways to install the LPAR tool for each supported operating system: installing a package or installing a tar file.

Recommended is the installation of the package for the appropriate operating system. This requires administration rights on the system where the LPAR tool is to be installed. The LPAR tool is therefore available to all users on the system. Of course, a user still needs a login on the HMCs.

The possibility of using a tar file for the installation is intended in the case that the user does not have administration rights. The tar file can then be unpacked by the user in his home directory. The LPAR tool is then only available to this user. Of course, other users on the same system can also install and use the LPAR tool via tar in their home directory.

3. Installation on AIX

A package is available for installation on AIX: *pwrcmps.lpar.1.X.X.X.bff*. This contains the fileset *pwrcmps.lpar.rte*. The package can be downloaded from the website *https://powercampus.de/download*.

On the AIX system, the installation must be performed with root privileges:

```
# installp -acXYd pwrcmps.lpar.1.X.X.X.bff all
+-----------------------------------------------------------------------------+
                    Pre-installation Verification...
+-----------------------------------------------------------------------------+
Verifying selections...done
Verifying requisites...done
Results...
```

```
SUCCESSES
---------
  Filesets listed in this section passed pre-installation verification
  and will be installed.

  Selected Filesets
  -----------------
  pwrcmps.lpar.rte 1.1.0.0                  # LPAR-Tool Runtime Environment

  << End of Success Section >>

+-----------------------------------------------------------------------------+
                    BUILDDATE Verification ...
+-----------------------------------------------------------------------------+
Verifying build dates...done
FILESET STATISTICS
------------------
    1  Selected to be installed, of which:
        1  Passed pre-installation verification
  ----
    1  Total to be installed

+-----------------------------------------------------------------------------+
                         Installing Software...
+-----------------------------------------------------------------------------+

installp: APPLYING software for:
       pwrcmps.lpar.rte 1.1.0.0

Finished processing all filesets.  (Total time:  1 secs).

+-----------------------------------------------------------------------------+
                                Summaries:
+-----------------------------------------------------------------------------+

Installation Summary
--------------------
Name                        Level           Part        Event       Result
-------------------------------------------------------------------------------
pwrcmps.lpar.rte            1.1.0.0         USR         APPLY       SUCCESS
```

The files of the fileset are installed under */opt/pwrcmps*. The programs (*hmc*, *ms*, *lpar*, and *vios*) are located at */opt/pwrcmps/bin*, a sample configuration file (*sample.lpar.cfg*) under */opt/pwrcmps/etc*, and a sample license file (*sample.lpar.lic*).

4. Installation on Linux

For installation under Linux, an RPM package is available: *pwrcmps-lpar-rte-1.X.X-X.rpm*. The RPM package can be downloaded from the website *https://powercampus.de/download*.

```
# rpm -i pwrcmps-lpar-rte-1.X.X-X.rpm
…
```

The files of the package are installed under */opt/pwrcmps*. The binaries (*hmc*, *ms*, *lpar*, and *vios*) are located at */opt/pwrcmps/bin*, a sample configuration file (*sample.lpar.cfg*) at */opt/pwrcmps/etc*, and a sample license file.

5. Installation on MacOS

The installation under MacOS is still in preparation.

6. Installation tar-File

If it is not possible or not desired to install the packages for the corresponding OS derivative, the LPAR tool can also be installed by unzipping a tar file.

The necessary tar file can be downloaded from the website https://powercampus.de/download. The tar file should be unpacked in the home directory.

```
$ pwd
/home/user01
$ tar xvf pwrcmps.lpar.0.1.1.aix.tar
x hmc, 434625 bytes, 849 tape blocks
x ms, 400955 bytes, 784 tape blocks
x lpar, 451557 bytes, 882 tape blocks
x vios, 405458 bytes, 792 tape blocks
x sample.lpar.cfg, 223 bytes, 1 tape blocks
x sample.lpar.lic, 184 bytes, 1 tape blocks
$
```

The binaries (*hmc*, *ms*, *lpar* and *vios*), as well as a sample configuration file (*sample.lpar.cfg*) and a sample license file (*sample.lpar.lic*) are then directly in the home directory. The binaries can be moved to any other location, e.g. ~/*bin*. The configuration file *.lpar.cfg* must be in the home directory. The license file can in principle be at any position, but the default is the home directory.

7. Configuration of the LPAR tool

Installing the LPAR tool as a package under */opt/pwrcmps* creates a sample configuration file */opt/pwrcmps/etc/sample.lpar.cfg* with the following content:

```
$ cat /opt/pwrcmps/etc/lpar.cfg
# Directory where the files hmc.list, ms.list and lpar.list are stored
# Default: ~
#ConfigDirectory ~

# Where to find the license file. Default: /opt/pwrcmps/etc/lpar.lic
#LicenseFile /opt/pwrcmps/etc/lpar.lic

# How long (minutes) inactive master connection should persist. Default: 600
#ControlPersist 600

# Time interval within server is expected to send alive message.
# Default: 10 seconds
#ServerAliveInterval 10

# Maximum number of outstanding alive messages from master.
# Default: 2
#ServerAliveCountMax 2

# The lowest virtual slot for a client adapter.
# Default: 10
```

```
#LowestVirtualClientSlot 10

# The lowest virtual slot for a server adapter.
# Default: 20
# LowestVirtualServerSlot 20
```

If the configuration of the LPAR tool is to be adapted system-wide, the sample configuration file can be copied to */opt/pwrcmps/etc/lpar.cfg* and edited. In many cases, the default configuration is already suitable. Each user of the LPAR tool can have a different configuration via the file *.lpar.cfg* in his home directory. The user-specific configuration file overwrites the settings from the system-wide configuration file.

The LPAR tool stores the registered HMCs, managed systems and LPARs in the 3 files *hmc.list*, *ms.list* and *lpar.list*. The *ConfigDirectory* parameter can be used to configure where these files should be stored. The default setting is "~", the respective home directory.

The LPAR tool requires a valid license key. The parameter *LicenseFile* can be used to specify in which file the license is entered. The default is */opt/pwrcmps/etc/lpar.lic* or *~/.lpar.lic* (home directory of the user) and should not be overwritten.

The LPAR tool uses SSH-Master-Connections, with the parameter *ControlPersist* you can specify how long this connection should last (in seconds). The default value is *5* minutes.

The lowest virtual slot number of virtual client adapters can be configured using the *LowestVirtualClientSlot* parameter. This parameter is used when the LPAR tool generates the slot number for a client adapter. The default value is *10*.

The lowest virtual slot number of virtual server adapters can be configured using the *LowestVirtualServerSlot* parameter. This parameter is used when the LPAR tool generates the slot number for a server adapter on a Virtual I/O Server. The default value is *20*.

8. Installation of the License

A license is required to use the LPAR tool. A 30-day trial license is available via *info@powercampus.de*.

If the LPAR tool was installed from a package, the license should be entered in the file */opt/pwrcmps/etc/lpar.lic*.

If the LPAR tool was installed from a tar file, the license can in principle be installed at any point, but the path must then be configured in the user's home directory via the configuration file *.lpar.cfg*.

Generating a license requires the serial numbers of the HMCs used. The serial number can be obtained by logging into the HMC using ssh and the command "*lshmc -v*":

```
$ ssh hmc01
hmc01> lshmc -v
…
*SE 123ABCD
…
hmc01> exit
$
```

(The serial number is in the line beginning with "* *SE*".)

The license key must be requested from *PowerCampus*[01]. Usually, the license key is sent by e-mail. The license key from the mail must then be entered in the license file:

```
# vi /opt/pwrcmps/etc/lpar.lic

HMCs: 123456A,123456B
HMCs: 123456C,123456D
ExpirationDate: 01.08.2018
LicenseKey: f6a3c57bf5c4d1a03c4ddf3ac0354753

#
```

Or in case of installation in a home directory:

```
$ vi ~/.lpar.lic

HMCs: 123456A,123456B
HMCs: 123456C,123456D
ExpirationDate: 01.08.2018
LicenseKey: f6a3c57bf5c4d1a03c4ddf3ac0354753

$
```

2. Using the LPAR tool

1. Configuration of OpenSSH

The LPAR tool uses *ssh* to access the registered HMCs. For the LPAR tool to work, on each HMC an account is required that can be used by *ssh*. By default it is assumed that the account on the HMCs uses the same username as on the local system. Any other username is also possible. In order not having to enter passwords or passphrases constantly, a public key should be generated and stored on the HMC:

```
$ cat .ssh/id_rsa.pub
ssh-rsa XYS…

$ ssh HMC
…
HMC> mkauthkeys -a „ssh-rsa XYS…“
```

Now that the public key is stored on the HMC, an ssh-agent should be started:

```
$ eval $( ssh-agent )
$ ssh-add
<Passphrase>
```

2. Registring an HMC

Managed Systems and LPARs to be administered with the LPAR tool must be known to the LPAR tool. For this it is necessary to register the corresponding HMC(s). The command "*hmc add*" registers the specified HMC as well as all managed systems connected to the HMC and their LPARs.

```
$ hmc add hmc01
ms05
ms01
ms03
ms02
ms04
$
```

The output shows the newly registered Managed Systems. The registered HMCs can be listed with the command "*hmc list*" or "*hmc show*":

```
$ hmc list
hmc01
$ hmc show
HMC       SERIAL    MODEL
hmc01     123XXX    7042-CR9
$
```

If the user name on the local system does not match the user name on the HMC, the HMC user name must be specified with the option "*-u*" when registering:

```
$ hmc add -u hscroot hmc02
ms11
ms12
$
```

In this way, all HMCs can be registered, at least as far as licensing permits.

If a managed system is connected to two HMCs, which is likely to be the case in practice, the second HMC should also be registered. This has the advantage that in case of unavailability of one HMC, the other HMC is automatically used by the LPAR. This is transparent to the user.

Of course you can de-register an HMC again, this can be done by the command "*hmc remove*":

```
$ hmc show
HMC      SERIAL   MODEL
hmc01    XXXXXX   7042-CR9
hmc02    XXXXXX   7042-CR7
$ hmc remove hmc01
ms11
ms12
$ hmc show
HMC      SERIAL   MODEL
hmc02    XXXXXX   7042-CR7
$
```

The information which LPAR belongs to which managed system and which managed system belongs to which HMC is stored in 3 files: *hmc.list*, *ms.list* and *lpar.list*. The LPAR tool reads these files to know the mapping. If a new LPAR is now created using the HMC GUI, or if an LPAR is moved using the HMC GUI to another managed system, the saved mappings are not complete and correct anymore. In this case you can re-read the mappings by contacting all HMCs and querying the list of Managed Systems and LPARs. This is done by the command "*hmc rescan*":

```
$ hmc rescan
ms03
ms01
ms05
ms04
ms02
ms11
ms12
$
```

After a run of "*hmc rescan*" the mappings are correct again.

3. Overview of the Commands

All 4 commands of the LPAR tool work in a similar way. Each of the commands expects the specification of a keyword. The keyword determines the functionality to perform. Before and after the keyword, supported options can be specified. The remaining arguments are used by the function to be performed:

```
command [options] keyword [options] arguments
```

Each of the 4 commands supports the keywords "*list*" and "*show*". These two keywords only access the internal map files, so there is no SSH access to the HMCs. The keyword „*list*" lists only the names of registered HMCs, Managed Systems, LPARs, or Virtual I / O Servers. The keyword "*show*" returns some additional information. Here are some examples:

```
$ hmc list
hmc01
hmc02
$ ms show
MS    SERIAL   MODEL     HMCs
ms01  XXXXXXX  8203-E4A  hmc01,hmc02
ms02  XXXXXXX  8233-E8B  hmc01
ms03  XXXXXXX  8406-71Y  hmc02
ms04  XXXXXXX  8284-22A  hmc02
…
$ lpar list
lpar1
lpar2
lpar3
…
$ vios show
VIOS        ID   SERIAL    TYPE       MS
ms01-vio1   1    XXXXXXXX  vioserver  ms01
ms01-vio2   2    XXXXXXXX  vioserver  ms01
ms02-vio1   1    XXXXXXXX  vioserver  ms02
…
$
```

In all cases you can limit the output by using arguments:

```
$ hmc list hmc01 hmc02
hmc01
hmc02
$ hmc show hmc02
HMC       SERIAL   MODEL
hmc02     XXXXXX   7042-CR7
$
```

The commands *lpar* and *vios* support in addition the option *-m*, with which a managed system can be specified and the option *-h*, with which an HMC can be specified. Only the LPARs or virtual I/O servers of the specified managed system or HMC will be shown:

```
$ lpar -m ms01 list
lpar1
lpar2
ms01-vio1
ms01-vio2
$ vios -m ms01 show
VIOS        ID   SERIAL    TYPE       MS
ms01-vio1   1    XXXXXXXX  vioserver  ms01
ms01-vio2   2    XXXXXXXX  vioserver  ms01
$
```

If you want to see the possible keywords for one of the commands, you can call the command with the keyword "*help*" and the argument "*usage*":

```
$ vios help usage
   Usage: vios [option …] keyword [arg …]
          vios -V
Recognized keywords:
   [-h <hmc>] [-m <ms>] chrep [-v] <vios> <additional-size>
   [-h <hmc>] -m <ms> [-p <profile>] create [-i <lparid] [-v] <vios>
   [-h <hmc>] [-m <ms>] list [<vios> …]
   [-h <hmc>] [-m <ms>] loadopt [-f] [-r] [-v] <vios> <vios> <vtopt> <media>
…
$
```

Alternatively, you can simply call the command without a keyword. An error message will be displayed, as well as a list of valid keywords:

```
$ lpar
ERROR: missing keyword
   Usage: lpar [option …] keyword [arg …]
          lpar -V
Recognized keywords:
   [-h <hmc>] [-m <ms>] [-p <profile>] activate [-b norm|dd|ds|of|sms] [-c] [-v] <lpar>
…
$
```

Options can generally be specified either before or after the keyword, so the following commands are equivalent:

```
$ lpar -p standard -b sms activate lpar1
oder
$ lpar -p standard activate -b sms lpar1
oder
$ lpar activate -p standard -b sms lpar1
oder
$ lpar -b sms activate -p standard lpar1
$
```

All keywords (except *help*, *list*, and *show*) support the *-v* option. This option stands for *verbose-only*. This means that the specified function is not executed, but the commands that would be executed on the HMC are displayed:

```
$ lpar -v -c addfc lpar1 11 ms01-vio1 105
hmc01: chhwres -m ms01 -r virtualio –rsubtype fc -o a -p lpar1 -s 11 -a
adapter_type=client,remote_lpar_name=ms01-vio1,remote_slot_num=105
hmc01: lssyscfg -m ms01 -r lpar –filter lpar_names=lpar1 -F curr_profile
hmc01: chsyscfg -m ms01 -r prof -i
lpar_name=lpar1,name=standard,“virtual_fc_adapters+=„“11/client//ms01-vio1/105//0“““
$
```

In addition to the HMC-CLI commands, the HMC where the command or commands would be executed are shown. This is useful if you want to see what commands are finally executed on the HMC command line.

The version of the installed LPAR tool can be displayed by specifying the option "*-V*" with each of the 4 commands:

```
$ ms -V
```

```
Version: 1.2.0
(c) 2017-2018 by PowerCampus
$
```

4. Using the keyword *help*

All LPAR-Tool commands return a usage message when called without arguments. This contains an overview of all available keywords:

```
$ lpar
ERROR: missing keyword
   Usage: lpar [option …] keyword [arg …]
          lpar -V
Recognized keywords:
   [-h <hmc>] [-m <ms>] [-p <profile>] activate [-b norm|dd|ds|of|sms] [-c] [-v] <lpar>
…
$
```

If one of the commands is called with the keyword '*help*', the functionality of the help keyword is displayed.

```
$ lpar help
Help is available for the following topics:
   lpar help blueprint | console | dlpar | mobility | operations | profiles | usage
   lpar help eth | fc | mem | proc | scsi | serial

Specific help is also available for each of the supported keywords:
   lpar help <keyword>

For a complete list of all keywords try:
   lpar help usage
$
```

If you enter one of the specified topics as an argument for the keyword '*help*', all keywords related to that keyword will be listed. This makes it easy to see which commands are available for topic. For example, if you want to know which keywords exist in connection with the console, you can use '*lpar help console*':

```
$ lpar help console
   Usage: lpar [option ...] keyword [arg ...]
          lpar -V
Recognized keywords:
   [-h <hmc>] [-m <ms>] console|mkvterm [-f] [-v] <lpar>
   [-h <hmc>] [-m <ms>] rmconsole|rmvterm [-v] <lpar>

For more specific help try
   lpar help <keyword>
with one of the above keywords
$
```

If you want more detailed information about options or arguments of a keyword, the corresponding keyword can be given as an argument for *help*, here e.g. the keyword *chmem* of the command *lpar*:

```
$ lpar help chmem
Usage:
```

```
   lpar [-h <hmc>] [-m <ms>] [-p <profile>] chmem [-v] <lpar> [<desired>|<desired> <max>|
<min> <desired> <max>] [<attributes>]
   lpar [-h <hmc>] [-m <ms>] chmem -d [-v] <lpar> [<desired>] [<attributes>]
valid attributes are:
   mem_weight
   mem_expansion
      0 - disable Active Memory Expansion (only in profile possible)
      1.00-10.00 - expansion factor
   hardware_mem_encryption
      0 - disable hardware-accelerated encryption
      1 - enable hardware-accelerated encryption
   hardware_mem_expansion
      0 - disable hardware-accelerated Active Memory Expansion
      1 - enable hardware-accelerated Active Memory Expansion

EXAMPLES:
Set memory to 1024 MB (DLPAR and current profile):
   lpar chmem lpar1 1024
Set memory to 2048 MB (DLPAR only):
   lpar chmem -d lpar1 2048
Set memory expansion factor to 1.4 and disable hardware-accelerated encryption (in profile
standard):
   lpar -p standard chmem lpar1 mem_expansion=1.4 hardware_mem_encryption=0
Set memory to minimum 1024, desired 2048 and maximum 4096 (in profile standard only):
   lpar -p standard chmem lpar1 1024 2048 4096
$
```

In addition to the options and arguments, any available attributes are listed and explained, as well as some examples showing the use of the keyword.

Especially at the beginning of using the LPAR tool, the large number of functions and keywords can surprise the user and make finding the desired functionality difficult. The possibility of a topic-specific listing, as shown above (example console) can be enormously helpful here. In addition, there is the possibility to adapt the help to your own needs. For this you can configure in the configuration file for each of the 4 commands which keywords are displayed. This allows to limit the overview to the already known keywords. As an example, we show this for the command *lpar*: in the configuration file */opt/pwrcmps/etc/lpar.cfg* or *~/.lpar.cfg* the parameter *LparUsageShow* can be configured with a list of the keywords that should be displayed:

```
$ vi ~/.lpar.cfg
…
LparUsageShow list,show,status,vslots,console
```

In the example shown, only the keywords *list*, *show*, *status*, *vslots*, and *console* are shown. All help and usage output from the command only show those keywords:

```
$ lpar
ERROR: missing keyword
   Usage: lpar [option ...] keyword [arg ...]
          lpar -V
Recognized keywords:
   [-h <hmc>] [-m <ms>] console|mkvterm [-f] [-v] <lpar>
   help [<keyword>|blueprint|console|dlpar|mobility|operations|profiles]
   [-h <hmc>] [-m <ms>] list [<lpar> ...]
   [-h <hmc>] [-m <ms>] show [<lpar> ...]
   [-h <hmc>] [-m <ms>] stat|status [-v] [<lpar> ...]
   [-h <hmc>] [-m <ms>] [-p <profile>] vslots [-v] <lpar>
```

```
$
```

The outputs are thus shorter and faster manageable. Over time, you can then gradually add more keywords to this list.

For the 3 other commands there are corresponding parameters *HmcUsageShow*, *MsUsageShow* and *ViosUsageShow*.

3. Administration of LPARs

First, we'll show some simpler, but more common, operations on LPARs using the LPAR tool:

- Status of an LPAR
- Configuration of an LPAR
- Activating an LPAR
- Shuting down an LPAR
- Opening the console for an LPAR

1. Status of an LPAR

To display the status of an LPAR there is the command "*lpar status*" or shorter "*lpar stat*". This command can also list the status of several or even all LPARs with one command.

List the status of an LPAR:

```
$ lpar stat lpar1
NAME    ID  TYPE           STATUS   PROFILE     RMC   PROCS PROCUNITS  MEMORY  OS
lpar1   20  aixlinux       Running  standard  active    2      0.4     4096   AIX 7.1
7100-04-05-1720
$
```

List status of multiple LPARs:

```
$ lpar stat lpar1 lpar2 lpar3
NAME    ID  TYPE           STATUS   PROFILE     RMC   PROCS PROCUNITS  MEMORY  OS
lpar1   20  aixlinux       Running  standard  active    2      0.4     4096   AIX 7.1
7100-04-05-1720
lpar2   16  aixlinux       Running  standard  active    1      0.4     4096   AIX 7.1
7100-04-05-1720
lpar3   13  aixlinux  Not Activated        -  inactive  0        -        0  Unknown
$
```

List status of all LPARs of a managed system:

```
$ lpar -m ms01 stat
NAME    ID  TYPE           STATUS   PROFILE     RMC   PROCS PROCUNITS  MEMORY  OS
lpar1   20  aixlinux       Running  standard  active    2      0.4     4096   AIX 7.1
7100-04-05-1720
lpar2   16  aixlinux       Running  standard  active    1      0.4     4096   AIX 7.1
7100-04-05-1720
lpar3   13  aixlinux  Not Activated        -  inactive  0        -        0  Unknown
lpar4   13  aixlinux  Not Activated        -  inactive  0        -        0  Unknown
lpar5   13  aixlinux  Not Activated        -  inactive  0        -        0  Unknown
$
```

List status of all LPARs:

```
$ lpar stat
NAME    ID  TYPE            STATUS   PROFILE     RMC   PROCS PROCUNITS  MEMORY  OS
lpar1   20  aixlinux        Running  standard  active    2      0.4     4096   AIX 7.1
7100-04-05-1720
lpar2   16  aixlinux        Running  standard  active    1      0.4     4096   AIX 7.1
7100-04-05-1720
lpar3   13  aixlinux  Not Activated         -  inactive  0        -        0  Unknown
…
$
```

In addition to the status of the LPAR, the status of the RMC connection and other information is also displayed.

2. Properties of an LPAR

The current properties of an LPAR can be most easily displayed with the command "*lpar properties*" or shorter "*lpar prop*":

```
$ lpar prop lpar1
                                      PROFILE                PROC_COMPAT_MODE
NAME     ID  SERIAL   LPAR_ENV  DEFAULT   CURRENT   SYNC  DESIRED  CURRENT  BOOT_MODE
KEYLOCK
lpar1    4  XXXXXXXX  aixlinux  standard  standard   0    default  POWER7    norm
norm
$
```

Some properties are not set by profile but apply to an LPAR regardless of the profile used. Examples of such properties are the LPAR name, the default profile name, or the remote_restart_capable property. The properties can be changed by using the command "*lpar chproperties*" or shorter "*lpar chprop*", shown here by the example of the property *new_name*:

```
$ lpar chprop lpar1 new_name=lpar100
$
```

The above command renamed the LPAR. However, the name change does not occur in the local map files, so the LPAR is still known under the old name, but can not be addressed by the old name:

```
$ lpar show lpar1
       MEMORY          MEMORY         HUGEPAGES
NAME    MODE  AME  MIN  CURR  MAX  MIN  CURR  MAX
ERROR: lparMemory(): remote HMC command returned an error (1)
CMD on hmc01: lshwres -m ms01 -r mem —level lpar —filter lpar_names=lpar1
StdErr: HSCL8012 The partition named lpar1 was not found. Please check your entry and retry
the command.
-   -  -  -  -  -  -  -  -
$
```

An "*hmc rescan*" updates the mapping files again and then the renamed LPAR can now be addressed by its new name.

However, to rename an LPAR, the simpler way is to use the "*lpar rename*" command, which also updates the mapping files:

```
$ lpar rename lpar100 lpar1
$
```

As another example, we'll show the change of the *sync_curr_profile* property, which indicates whether the current configuration of an LPAR should automatically be synchronized with the current active profile. If this is activated, changes to the LPAR are automatically made in the profile as well. Current configuration and profile are then always synchronous.

```
$ lpar chprop lpar1 sync_curr_profile=1
$
```

Valid values for the *sync_curr_profile* property are:

```
0 - Synchronisierung deaktivieren
1 - Synchronisierung aktivieren
2 - Synchronisierung vorübergehend aussetzen (suspend), bis das nächste Mal ein Profil
aktiviert oder angewendet wird
```

If synchronization is activated, the currently active profile can no longer be changed:

```
$ lpar -p standard chmem lpar1 mem_expansion=1.4
ERROR: lparSetMemoryProfile(): remote HMC command returned an error (1)
CMD on hmc01: chsyscfg -m ms01 -r prof -i lpar_name=lpar1,name=standard,mem_expansion=1.4
StdErr: An error occurred while changing the partition profile named standard.
StdErr: This profile is synchronized with the partition's current configuration. To update
this profile, you can specify the force option on this command, or you can turn off current
profile synchronization for the partition. If you specify the force option on this command,
this profile will be updated and synchronization of this profile will be suspended until
the next time this profile is activated or applied.
$
```

3. Aktivating an LPAR

If an LPAR is in the "*Not Activated*" state, it can be reactivated with the "*lpar activate*" command. In the simplest case this works as follows:

```
$ lpar activate lpar1
$
```

The LPAR is activated with the current profile. Depending on the configuration of the profile, the LPAR is automatically booted. Which of the profiles is the current profile can be determined with the help of "*lpar properties*" or shorter "*lpar prop*":

```
$ lpar prop lpar1
                                        PROFILE             PROC_COMPAT_MODE
```

```
NAME     ID  SERIAL   LPAR_ENV  DEFAULT   CURRENT   SYNC  DESIRED  CURRENT  BOOT_MODE
KEYLOCK
lpar1    4  XXXXXXXX  aixlinux  standard  standard   0    default  POWER7    norm
norm
$
```

You can of course display also the „*properties*“ from several LPARs:

```
$ lpar -m ms09 properties
                                        PROFILE              PROC_COMPAT_MODE
NAME     ID  SERIAL   LPAR_ENV  DEFAULT   CURRENT   SYNC  DESIRED  CURRENT  BOOT_MODE
KEYLOCK
lpar1    4  XXXXXXXX  aixlinux  standard  standard   0    default  POWER7    norm
norm
lpar2    5  XXXXXXXX  aixlinux  standard  standard   0    default  POWER7    norm
norm
lpar3    6  XXXXXXXX  aixlinux  standard  standard   0    default  POWER7    norm
norm
lpar4    7  XXXXXXXX  aixlinux  standard  standard   0    default  POWER7    norm
norm
lpar5    9  XXXXXXXX  aixlinux  standard  standard   0    default  POWER7    norm
norm
$
```

When activating an LPAR for the first time, a profile must be specified, the LPAR does not yet have a current profile:

```
$ lpar -m ms09 create
Creating LPAR lpar6:
done
Register LPAR
done
$ lpar prop lpar6
                                        PROFILE              PROC_COMPAT_MODE
NAME     ID  SERIAL   LPAR_ENV  DEFAULT   CURRENT   SYNC  DESIRED  CURRENT  BOOT_MODE
KEYLOCK
lpar6    8  XXXXXXXX  aixlinux  standard      -      0    default  POWER7    norm
norm
$
```

If you attempt to start the LPAR without specifying a profile, you will get the following error message:

```
$ lpar activate -b sms lpar6
ERROR: lparActivate(): remote HMC command returned an error (1)
CMD on hmc01: chsysstate -m ms09 -r lpar -o on -n lpar6 -b sms
StdErr: HSCL3680 Partition lpar6 cannot be activated due to insufficient resources in its
current configuration.  Please activate the partition with a profile.
$
```

The available profiles of an LPAR can be displayed with the command "*lpar lsprof*":

```
$ lpar lsprof lpar6
NAME             PROFILES
lpar6            standard
$
```

We are now activating the above LPAR with the profile '*standard*':

```
$ lpar activate -p standard -b sms lpar6
$
```

Now also a current profile for the LPAR is shown :

```
$ lpar prop lpar6
                                        PROFILE                PROC_COMPAT_MODE
NAME      ID  SERIAL    LPAR_ENV   DEFAULT    CURRENT    SYNC  DESIRED  CURRENT   BOOT_MODE
KEYLOCK
lpar6     8   XXXXXXXX  aixlinux   standard   standard    0    default  POWER7     norm
norm
$
```

If the LPAR is not to be booted automatically, the desired boot mode can also be specified:

```
$ lpar activate -b sms lpar1
$
```

The LPAR is activated and boots into the SMS menu. Alternatively, you can specify "*norm*" for normal *bootmode* or "*of*" for OpenFirmware.

When activating an LPAR, the desired profile can also be specified:

```
$ lpar -p standard activate -b of lpar1
$
```

Relatively often you will open a console after activating an LPAR. By using the option "*-c*" with "*lpar activate*" will open the console, so the command "*lpar console*" does not need to be executed:

```
$ lpar activate -c lpar1

 Open in progress

 Open completed.

IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM IBM
…
```

Further options can be displayed via the help:

```
$ lpar help activate
USAGE:
   lpar [-h <hmc>] [-m <ms>] [-p <profile>] activate [-b norm|dd|ds|of|sms] [-c] [-v]
<lpar>
   -b : the boot-mode to use
      norm - normal boot
      dd - diagnostic boot with default bootlist
      ds - diagnostic boot with stored bootlist
      of - boot to Open Firmware
      sms - boot to System Management Service
   -c : open console
   -p : activate LPAR using the specified profile

EXAMPLES:
```

```
Activate LPAR using the current profile:
   lpar activate lpar1
Activate LPAR using the current profile, boot into SMS and open a console:
   lpar activate -b sms -c lpar1
Activate LPAR with the profile named 'standard':
   lpar -p standard activate lpar1
$
```

4. Shutting down an LPAR

An LPAR should normally be shut down by the operating system of the LPAR, in the case of AIX with the command *shutdown* for example. Alternatively, this can also be done with the LPAR tool:

```
$ lpar shutdown lpar1
$
```

The operating system of the LPAR is signaled that it should shut down. The operating system will then shut down the system and the LPAR will be turned off ("*Not Activated*" state). In cases where the operating system stops responding (due to errors) or there is no operating system at all, the option "*-f*" (*force*) can be used, the LPAR is then simply turned off:

```
$ lpar shutdown -f lpar1
$
```

5. Console for an LPAR

A frequently used feature of the LPAR tool is the ability to launch easily a console for an LPAR at any time:

```
$ lpar console lpar1

 Open in progress

 Open completed.

 PowerPC Firmware
 Version AL720_121
 SMS 1.7 © Copyright IBM Corp. 2000,2008 All rights reserved.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 Main Menu
 1.   Select Language
 2.   Setup Remote IPL (Initial Program Load)
 3.   Change SCSI Settings
 4.   Select Console
 5.   Select Boot Options

…
```

Of course, for one LPAR, only one console can be open at one time. If a console is already open, you will get the following error message:

```
$ lpar console lpar1
```

```
 A terminal session is already open for this partition.
 Only one open session is allowed for a partition.
 Exiting….    Received end of file, Exiting.
               Shared connection to hmc01 closed.
$
```

By using the option "*-f*" (*force*) a console can be forced, the already open console is terminated:

```
$ lpar console -f lpar1


 Open in progress

 Open completed.

…
```

If the console is to be closed, the escape sequence "~." must be used:

```
 …
~.

 Terminate session? [y/n] y

Shared connection to hmc01 closed.
$
```

If a hanging console session is to be terminated, this can be done with the command "*lpar rmconsole*" or "*lpar rmvterm*":

```
$ lpar rmconsole lpar3
/bin/stty: standard input: Inappropriate ioctl for device
$
```

In the console session that terminates, the following message appears:

```

Connection has closed


This session is no longer connected. Please close this window.
```

Whether there are open consoles, can be checked indirectly. For this you can use the command "*hmc lslogon*" on the connected HMCs to list the running sessions:

```
$ hmc lslogon hmc01
USER_NAME   TTY_ID  LOGON_TIME        ACCESS_LOCATION
   TASK_NAME  TTY_ID  START_TIME  USER_NAME  PID
-          -       -                  -
   -          -       -           -          -
$
$ hmc lslogon hmc02
USER_NAME   TTY_ID  LOGON_TIME        ACCESS_LOCATION
   TASK_NAME  TTY_ID  START_TIME            USER_NAME  PID
```

```
ize0h29     pts/1   2018-10-10 09:38  172.20.132.167
   mkvterm    pts/1   Oct 10 09:38:18 2018  root       24583
$
```

A console session can be recognized by the task name *mkvterm*.

6. Live Partition Mobility (LPM)

Here we show, how to perform LPM with the help of the LPAR tool.

First we check the status of the LPAR *lpar2* and the status of the RMC connection. The easiest way is the command "*lpar stat*":

```
$ lpar status lpar2
NAME       ID       TYPE         STATUS          PROFILE        RMC   PROCS   PROCUNITS
MEMORY  OS
lpar2      5   aixlinux        Running         standard     active    1        0.1    4096
AIX 7.1 7100-04-06-1806
$
```

The LPAR is in the status „*Running*“ and the RMC connection is *active*. The LPAR is active with one processor core and 0.1 processor units, and it has 4 GB of main memory.

Next we check the processor compatibility mode:

```
$ lpar prop lpar2
                                        PROFILE                PROC_COMPAT_MODE
NAME     ID  SERIAL    LPAR_ENV   DEFAULT    CURRENT    SYNC  DESIRED  CURRENT   BOOT_MODE
KEYLOCK
lpar2    5  XXXXXXXX  aixlinux   standard   standard    0    default  POWER7     norm
norm
$
```

The LPAR is currently running in *POWER7* mode, i. the target platform must support at least *POWER7*. Finally, we check the VLAN and VSwitch used by the LPAR:

```
$ lpar vslots lpar2
SLOT  REQ  TYPE           DATA
0     1    serial/server  remote: -(any)/any status=unavailable hmc=1
1     1    serial/server  remote: -(any)/any status=unavailable hmc=1
5     0    eth            PVID=1234 VLANS=- XXXXXXXXXXXX ETHERNET0
10    0    fc/client      remote: ms01-vio1(1)/103 c050760XXXXX0052,c050760XXXXX0053
20    0    fc/client      remote: ms01-vio2(2)/203 c050760XXXXX0056,c050760XXXXX0057
$
```

The LPAR is using VLAN *1234* on VSwitch *ETHERNET0*.

Now we check the target-managed system *ms03*. It is an *S824*, a *POWER8* system. The available processor units can be displayed with "*ms procs*":

```
$ ms procs ms03
MS               INSTALLED  CONFIGURABLE  AVAILABLE
ms03               48.0          48.0        3.9
$
```

The available RAM can be displayed using "*ms mem*":

```
$ ms mem ms03
MS               INSTALLED   FIRMWARE    CONFIGURABLE  AVAILABLE
ms03             2097152     54272       2097152     753664
$
```

So there are enough resources available.

The VLAN *1234* on VSwitch *ETHERNET0* is also available, as the following command shows:

```
$ ms lsvswitch ms03
MS          VSWITCH          SWITCH_MODE  VLAN_IDS
ms03        ETHBLB           VEB          10,12,14
ms03        ETHERNET0(Default)  VEB       20,21,22,1234
$
```

Moving the LPAR to managed system *ms03* should be possible.

The manual checks we made above are also done when moving the LPAR from the HMC. It is also possible to carry out a so-called validation of the LPAR, during which it is checked whether all conditions for migration are met without actually moving the LPAR.

We now perform such a validation using the LPAR tool:

```
$ lpar validate lpar2 ms03

…

Warnings:
HSCLA4CD The management console cannot maintain the source Virtual I/O Server (VIOS) slot
number 203 for virtual fibre channel adapter 20 on the destination VIOS partition 1*MMMM-
TTT*SSSSSSS.
Shared connection to hmc01 closed.
$
```

The exit status of the validation is *0* despite all the messages:

```
$ echo $?
0
$
```

This means that the LPAR can be moved. We now perform the migration:

```
$ lpar migrate lpar2 ms03

…
```

```
Warnings:
HSCLA4CD The management console cannot maintain the source Virtual I/O Server (VIOS) slot
number 203 for virtual fibre channel adapter 20 on the destination VIOS partition 1*MMMM-
TTT*SSSSSSS.
Shared connection to hmc01 closed.
$
```

Only warnings were shown, which means the migration was successful. We'll take a quick look at where the LPAR is now:

```
$ lpar show lpar2
LPAR                 ID    SERIAL    TYPE        MS
lpar2                40    XXXXXXXXX aixlinux    ms03
$
```

This confirms that the migration was successful.

You can also move an inactive LPAR to another managed system. We will try this for the LPAR *lpar3* now:

```
$ lpar status lpar3
NAME      ID       TYPE         STATUS          PROFILE        RMC  PROCS  PROCUNITS
MEMORY  OS
lpar3      6   aixlinux  Not Activated         standard   inactive    1        0.3    2048
AIX 7.1 7100-04-05-1720
$
```

The LPAR also uses the VLAN *1234*. This time we don’t do a validation, instead we perform the migration immediately:

```
 $ lpar migrate lpar3 ms05

...

Warnings:
HSCLA295 As part of the migration process, the management console will create a new
migration profile containing the partition's current state.  The default is to use the
current profile, which will replace the existing definition of this profile.  While this
works for most scenarios, other options are possible.  You may specify a different existing
profile, which would be replaced with the current partition definition, or you may specify
a new profile to save the current partition state.
HSCLB505 The partition cannot use hardware-accelerated encryption on the destination
managed system because the destination managed system does not support hardware-accelerated
encryption.
HSCLB504 The migrating partition cannot use hardware-accelerated Active Memory Expansion on
the destination managed system because the destination managed system does not support
hardware-accelerated Active Memory Expansion.
HSCLA4CD The management console cannot maintain the source Virtual I/O Server (VIOS) slot
number 44 for virtual fibre channel adapter 10 on the destination VIOS partition 1*MMMM-
TTT*SSSSSSS.
Shared connection to hmc01 closed.
$
```

The LPAR has been moved to *ms05*:

```
$ lpar show lpar3
LPAR              ID    SERIAL    TYPE        MS
lpar3             39    XXXXXXXXX  aixlinux    ms05
$
```

4. Creation of LPARs

The LPAR tool makes it easy to create new LPARs. By using blueprint files even complex LPARs can be created with just one command.

1. Creation of a new LPAR

A new LPAR can be created with the command "*lpar create*". The managed system must be specified on which the LPAR is to be created:

```
$ lpar -m ms01 create
Creating LPAR lpar1:
done
Register LPAR
done
$
```

The LPAR is created without physical and virtual adapters. Since no profile name was specified, the default "*standard*" is used. The name of the LPAR is *lparN*, where *N* is counted up from *1*. Of course, the profile name and/or name of the LPAR can also be specified:

```
$ lpar -m ms01 -p myprofile create mylpar01
Creating LPAR mylpar01:
done
Register LPAR
done
$
```

The profile of the newly created LPAR can be displayed with the following commands:

- General properties: *lpar –p <profile> properties*
- Processor configuration: *lpar –p <profile> procs*
- Memory configuration: *lpar –p <profile> memory*
- Virtual I/O: *lpar –p <profile> vslots*
- Physical I/O: *lpar –p <profile> slots*

First, the general properties from the LPAR profile are shown (the keyword "*properties*" can be abbreviated to „*prop*"):

```
$ lpar -p myprofile prop mylpar01
                                                CONNECTION                ERR_PATH
ELECTRONIC_ERR
NAME             LPAR_ENV   AUTO_START  BOOT_MODE  MONITORING  WORK_GROUP  REPORTING
REPORTING
mylpar01         aixlinux         norm          1           0        none          0  null
$
```

It is possible to display properties from the profile of several LPARs simultaneously (as long as the profile name is the same), e.g.:

```
$ lpar -p standard -m ms01 prop
                                                  CONNECTION                ERR_PATH
ELECTRONIC_ERR
NAME              LPAR_ENV    AUTO_START   BOOT_MODE   MONITORING   WORK_GROUP   REPORTING
 REPORTING
lpar01            aixlinux          norm           1            0         none           0   null
lpar02            aixlinux          norm           0            0         none           0   null
lpar03            aixlinux          norm           0            0         none           0   null
ms01-vio1         vioserver         norm           0            0         none           0   -
ms01-vio2         vioserver         norm           1            0         none           0   -
mylpar01          aixlinux          norm           1            0         none           0   null
$
```

The subcommand "*lpar procs*" should be used to display the processor configuration, it works in the same way:

```
$ lpar -p myprofile procs mylpar01
                                PROCS           PROCUNITS                UNCAP
NAME              PROCMODE   MIN   CURR   MAX    MIN   CURR    MAX   SHARING   WEIGHT   POOL
mylpar01            shared     1      2     4    0.1    0.2    4.0     uncap        5   DefaultPool
$
```

Of course, any number of LPARs can be displayed at the same time, e.g. all LPARs connected to a specific HMC:

```
$ lpar -p standard -h hmc01 procs
                                PROCS           PROCUNITS                UNCAP
NAME              PROCMODE   MIN   CURR   MAX    MIN   CURR    MAX   SHARING   WEIGHT   POOL
lpar01              shared     1      1     1    0.1    0.4    1.0     uncap        5   DefaultPool
lpar02              shared     1      1     2    0.1    0.2    1.0     uncap      128   DefaultPool
lpar03              shared     1      6     8    0.1    3.0    8.0     uncap      128   DefaultPool
mylpar01            shared     1      2     4    0.1    0.2    4.0     uncap        5   DefaultPool
…
$
```

Similarly, one can display the memory configuration with the subcommand "*lpar memory*". The keyword "*memory*" can be abbreviated by "*mem*":

```
$ lpar -p myprofile mem mylpar01
                 MEMORY                    MEMORY                HUGEPAGES
NAME              MODE    AME       MIN      CURR       MAX   MIN   CURR   MAX
mylpar01            ded    0.0      1024      4096     16384   null   null   null
$
```

If an LPAR is active and you want to see the values of the active LPAR, just omit the profile specification. The commands then refer to the active configuration:

```
$ lpar mem mylpar01
                 MEMORY                    MEMORY                HUGEPAGES
NAME              MODE    AME       MIN      CURR       MAX   MIN   CURR   MAX
mylpar01            ded    0.0        0         0         0     0      0     0
$
```

Since the newly created LPAR *mylpar01* is not active, it currently requires no memory. LPARs that are currently active, of course, use memory:

```
$ lpar -m ms01 mem
                 MEMORY                    MEMORY                HUGEPAGES
NAME              MODE   AME       MIN      CURR       MAX   MIN   CURR   MAX
lpar01             ded   0.0      2048      4096     16384     0      0     0
lpar02             ded   0.0      2048      4096     16384     0      0     0
lpar03             ded   0.0      1024      4096     16384     0      0     0
mylpar01           ded   0.0         0         0         0     0      0     0
…
$
```

For the display of the virtual I/O configuration, there is the command "*lpar vslots*". Here, however, unlike the commands above, only one LPAR can be displayed at a time, because the output is much longer:

```
$ lpar -p myprofile vslots mylpar01
SLOT  REQ  TYPE           DATA
0     yes  serial/server  remote: (any)/any hmc=1
1     yes  serial/server  remote: (any)/any hmc=1
$
```

As expected, the newly created LPAR has only the two standard serial console adapters. If you omit the profile specification, the currently active configuration is shown.

The physical I/O configuration can be displayed with the command "*lpar slots*". Since the LPAR *mylpar01* has no physical I/O slots, no output is shown here.

On a second managed system, we create another LPAR with the same name *mylpar01*:

```
$ lpar create -m ms02 mylpar01
Creating LPAR mylpar01:
done
Register LPAR
Done
$
```

However, on a single managed system, the name of an LPAR must be unique:

```
$ lpar create -m ms02 mylpar01
Creating LPAR mylpar01:
ERROR: msLparCreate(): remote HMC command returned an error (1)
CMD on hmc01: mksyscfg -m ms02 -r lpar -i
name=mylpar01,profile_name=standard,lpar_env=aixlinux,max_virtual_slots=30,proc_mode=shared
,sharing_mode=uncap,uncap_weight=5,min_proc_units=0.1,desired_proc_units=0.2,max_proc_units
=4.0,min_procs=1,desired_procs=2,max_procs=4,min_mem=1024,desired_mem=4096,max_mem=16384
StdErr: An error occurred while creating the partition named mylpar01.
StdErr: HSCL05DE A partition in the managed system already uses the name mylpar01. Provide
another name for this partition.
```

The attempt to create a second LPAR with the same name fails, with the remark that there is already an LPAR of this name on the managed system.

In general, LPAR names should be unique throughout the environment so that it is always clear which LPAR is meant. The use of the LPAR tool becomes slightly more difficult if several LPARs have the same name. If you want to activate the LPAR, for example, then the specification of the LPAR name is no longer sufficient:

```
$ lpar activate mylpar01
ERROR: more than one LPAR matches specification
candidates are
  mylpar01 on ms01
  mylpar01 on ms02
$
```

There are two candidates here and it is not clear which of the two LPARs is meant. Of course, this can easily be clarified by specifying the managed system:

```
$ lpar -m ms01 activate -p myprofile mylpar01
$
```

2. Creating LPARs using Blueprints

Often, you want to create more complex LPARs, with virtual Ethernet adapters, virtual SCSI adapters, and virtual FC adapters. Here we briefly show the commands necessary to create an LPAR with a virtual Ethernet adapter (default VSwitch, PVID *1234*), two virtual FC adapters and two virtual SCSI adapters. For the slot numbers used, we have previously checked that these are available on the two virtual I/O servers. In addition, the LPAR should use between 1 and 16 GB of memory and start with 4 GB. It should be allowed to use between 1 and 8 processor cores and 0.1 to 2.0 processor units. When activated, it should start with 2 cores and 0.4 processor units.

```
$ lpar -m ms01 -p myprofile create mylpar01
Creating LPAR mylpar01:
done
Register LPAR
done
$ lpar -p myprofile addeth mylpar01 5 1234
$ lpar -p myprofile addfc mylpar01 10 ms01-vio1 55
$ lpar -p myprofile addfc mylpar01 20 ms01-vio2 55
$ lpar -p myprofile addscsi mylpar01 11 ms01-vio1 56
$ lpar -p myprofile addscsi mylpar01 21 ms01-vio2 56
$ lpar -p myprofile setmem mylpar01 1024 4096 16384
$ lpar -p myprofile setproc mylpar01 1 2 8
$ lpar -p myprofile setprocunits mylpar01 0.1 0.4 2.0
$
```

This would create the desired LPAR. By default, the LPAR tool creats the appropriate SCSI and FC server adapters on the two virtual I/O servers. This does not have to be done manually.

However, the virtual FC adapters still need to be mapped to physical FC adapters of the virtual I/O Server:

```
$ vios vfcmap ms01-vio1 vfchost5 fcs1
$ vios vfcmap ms01-vio2 vfchost5 fcs1
$
```

The correct *vfchost* devices can be easily found out with the help of the command "*vios npiv*", which is not shown here.

That's more than 10 commands. Of course you could write a short script executing the commands.

However, this effort is not necessary because the LPAR tool supports *blueprints* for generating LPARs.

A *blueprint* is simply a text file that defines the configuration of the LPARs to be created. You can easily create a *blueprint* with the help of an editor, here an example of a *blueprint*:

```
$ cat sample.bp
profile_name=standard
lpar_env=aixlinux
min_mem=1024
desired_mem=4096
max_mem=16384
proc_mode=shared
min_proc_units=0.1
desired_proc_units=0.4
max_proc_units=2.0
min_procs=1
desired_procs=2
max_procs=8
sharing_mode=uncap
uncap_weight=50
max_virtual_slots=30
# slot/is-IEEE/PVID/[VLANS]/[trunk-priority]/is-required
virtual_eth_adapters=5/0/1234//0/0
# slot/VIOS/min-remote-slot/is-required
virtual_fc_adapters=10/1st/50/0
virtual_fc_adapters=20/2nd/50/0
# slot/VIOS/min-remote-slot/is-required
virtual_scsi_adapters=11/1st/50/0
virtual_scsi_adapters=21/2nd/50/0
$
```

Comment lines start with '#'. Every other line has the format "*<attribute> = <value>*". The valid attributes are documented in the man page of "*mksyscfg*" on the HMC. The format of the "*virtual_fc_adapters*" and "*virtual_scsi_adapters*" attributes are as follows: First, the virtual slot number, then the virtual I/O server (where *1st* is the virtual I/O server with the lowest LPAR Id, *2nd* the next virtual I/O server and so on). The third parameter is the smallest allowed slot number on the virtual I/O server (starting from this slot number, the first unused slot number is searched for and then used). The last parameter specifies whether the adapter is required or not. The *blueprint* corresponds to the required configuration of our example.

The desired LPAR can now be created with just a single command:

```
$ lpar -m ms01 -p myprofile create -b sample.bp mylpar01
Creating LPAR mylpar01:
done
Register LPAR
done
Create virtual server adapters in profile of ms01-vio1
done
Create virtual fc server adapters for ms01-vio1
done
Create virtual scsi server adapters for ms01-vio1
done
```

```
Create virtual server adapters in profile of ms01-vio2
done
Create virtual fc server adapters for ms01-vio2
done
Create virtual scsi server adapters for ms01-vio2
done
$
```

On a test system this took about 30 seconds.

Creating a *blueprint* can be even easier. If you have an LPAR that is configured according to the desired specifications, it can be shown in the form of a *blueprint*:

```
$ lpar mkblueprint testlpar
name=standard
lpar_env=aixlinux
min_mem=1024
…
$
```

The output can easily be redirected to a file:

```
$ lpar mkblueprint testlpar >test.bp
$
```

If necessary, the *blueprint* file created in this way can be modified by means of an editor:

```
$ vi test.bp
$
```

The generated *blueprint* file can be used immediately to create a new LPAR:

```
$ lpar -m ms02 create -b test.bp
…
$
```

3. Deleting an LPAR

Deleting an LPAR is also easy with the LPAR tool. To delete an LPAR, there is the subcommand "*lpar delete*":

```
$ lpar -m ms01 delete mylpar01
Deleting LPAR mylpar01
ERROR: lparDelete(): remote HMC command returned an error (1)
CMD on hmc01: rmsyscfg -m ms01 -r lpar -n mylpar01
StdErr: An error occurred while deleting the partition named mylpar01.
StdErr: HSCL05E6 Partition mylpar01 delete failed. Cannot delete a partition when its state
is not in the Not Activated state. Perform a shutdown operation then delete the partition.
$
```

Of course, the LPAR to be deleted must not be active. This can be checked with "*lpar status*" or "*lpar stat*" before deleting:

```
$ lpar stat mylpar01
NAME              ID       TYPE         STATUS          PROFILE        RMC   PROCS
PROCUNITS  MEMORY  OS
mylpar01           3   aixlinux        Running         myprofile     active2        0.2
4096  Unknown
$
```

The LPAR can be shut down using the OS, or with the subcommand "*lpar shutdown*". The command "*lpar shutdown*" triggers an orderly shutdown of the LPAR by the HMC:

```
$ lpar shutdown -m ms01 mylpar01
$
```

If the status is "*Not Activated*", the LPAR can be deleted:

```
$ lpar stat mylpar01
NAME              ID       TYPE         STATUS          PROFILE        RMC   PROCS
PROCUNITS  MEMORY  OS
mylpar01           3   aixlinux  Not Activated        myprofile   inactive    2
0.2    4096  Unknown
$
```

Deleting the LPAR should now be successfully:

```
$ lpar -m ms09 delete mylpar01
Deleting LPAR mylpar01
done
ms09-vio1
Remove virtual fc server adapters on ms09-vio1
done
Remove virtual fc server adapters on ms09-vio1
done
ms09-vio2
Remove virtual fc server adapters on ms09-vio2
done
Remove virtual fc server adapters on ms09-vio2
Done
$
```

When the LPAR is deleted, any virtual adapters for the LPAR are also deleted on the virtual I/O servers, including the profiles of those virtual I/O Servers.

5. DLPAR-Operations

Physical and virtual resources can be dynamically added or removed during runtime of an LPAR while AIX or Linux are running. The only prerequisite for this is a working RMC connection to the HMC. The LPAR tool makes it easy to perform DLPAR operations. For all DLPAR operations with the LPAR tool, the current profile is also adjusted by default. On the GUI, this has to be done manually and will frequently be forgotten, which leads to problems after a new activation. Optionally, you can perform a DLPAR operation only at runtime, without adjusting the current profile. The option "*-d*" is used for this purpose. It is also possible to make the change only in a profile if e.g. the LPAR is disabled, in the SMS menu, or there is no active RMC connection between LPAR and HMC. The following subsections each show how DLPAR operations can be performed on different resources.

1. Changing the memory of an LPAR

A common task in the management of LPARs is the expansion of main memory. The profile of an LPAR indicates within which limits the main memory can be dynamically enlarged and reduced. These limits can be viewed in different ways:

```
$ lpar memory lpar1
         MEMORY               MEMORY            HUGEPAGES
NAME      MODE   AME     MIN    CURR     MAX   MIN  CURR  MAX
lpar1      ded   0.0    1024    4096    8192     0     0    0
$
```

The keyword "*memory*" can also be abbreviated with "*mem*".

The current configuration of the LPAR is displayed. If you want to display the configuration resulting from a profile, then the option "*-p <profile>*" must be used:

```
$ lpar -p standard mem lpar1
         MEMORY               MEMORY            HUGEPAGES
NAME      MODE   AME     MIN    CURR     MAX   MIN  CURR  MAX
lpar1      ded   0.0    1024    4096    8192     0     0    0
$
```

For our example LPAR *lpar1*, the memory limits are configured to be at least 1024 MB and at most 8192 MB. The current memory size is 4096 MB. The profile also has 4096 MB configured as the desired value.

To dynamically change the memory, an active RMC connection is needed. This can be easily found out with the command "*lpar status*":

```
$ lpar status lpar1
NAME       ID       TYPE      STATUS          PROFILE      RMC   PROCS  PROCUNITS  MEMORY
OS
lpar1      3    aixlinux     Running         standard   active    1        0.1    4096  AIX
7.1 7100-04-06-1806
$
```

The main memory of the LPAR *lpar1* should be extended by 1024 MB, both dynamically and in the current profile. We quickly check if 1024 MB RAM is available on the managed system. For this we need the associated managed system:

```
$ lpar show lpar1
LPAR      ID   SERIAL   TYPE      MS
lpar1     3    XXXXXXXX  aixlinux  ms01
$
```

The LPAR *lpar1* is located on the managed system *ms01*.

The managed system *ms01* still has sufficient free RAM capacity:

```
$ ms mem ms01
MS        INSTALLED   FIRMWARE   CONFIGURABLE   AVAILABLE
ms01      2097152     55552      2097152        717568
$
```

We now perform the RAM extension of *lpar1*:

```
$ lpar addmem lpar1 1024
$
```

The additional memory is immediately available to the LPAR. The profile of the LPAR will automatically be changed accordingly, so that the next time it is activated with the current profile, the same main memory size will be used.

If the memory size is only temporarily increased, the option "*-d*" (*dynamic only*) can be used. The old value is retained in the profile and the extension is only carried out dynamically:

```
$ lpar -d addmem lpar1 1024
$
```

If the LPAR is turned off and is activated later again, then it will start up with 2048 MB of RAM.

If you only want to change the memory size in the profile, the option "*-p <profile>*" is used instead of the option "*-d*":

```
$ lpar -p standard addmem lpar1 1024
$
```

If you want to reduce the memory of an LPAR, then there is the similar command "*lpar rmmem*" with the same options as "*lpar addmem*".

2. Changing main Memory Limits in the Profile

The minimum and maximum RAM size of an LPAR can only be changed in the profile. For the change to take effect, the LPAR must be stopped and then restarted using the changed profile.

If you want to increase *lpar1*'s memory to 16384 MB, this is not possible at runtime due to the maximum memory limit of 8192 MB. To change this, the profile must be adjusted, and this limit must be increased to a value of at least 16384 MB. Only then can the desired memory size of 16384 MB be configured. A change of the memory limits in the profile can be carried out with the command "*lpar chmem*":

```
$ lpar -p standard chmem lpar1 1024 16384 32768
$
```

In the example, the minimum value was left at 1024 MB, the desired memory size set to 16384 MB and the maximum value at 32768 MB. This leaves buffers for later increases, which can then be dynamically performed up to a size of 32768 MB.

The LPAR must be reactivated with the changed profile after shutting down and powering off:

```
$ lpar -p standard activate lpar1
$
```

3. Changing the Number of Processors and Processor Units

For performance bottlenecks, the number of processor cores and the number of processor units can be changed. The profile of an LPAR indicates within which limits these can be dynamically enlarged and reduced. These limits can be viewed in different ways:

```
$ lpar procs lpar1
                                  PROCS            PROCUNITS                UNCAP
NAME   PROCCOMPAT  PROCMODE  MIN  CURR  MAX   MIN  CURR   MAX  SHARING  WEIGHT  POOL
lpar1    POWER8      shared    1     1    2   0.1  0.1    0.4    uncap     100  DefaultPool
$
```

The current configuration of the LPAR is displayed. If you want to look at the configuration resulting from a profile, then the option "*-p <profile>*" must be used:

```
$ lpar -p standard procs lpar1
                                  PROCS            PROCUNITS                UNCAP
NAME   PROCCOMPAT  PROCMODE  MIN  CURR  MAX   MIN  CURR   MAX  SHARING  WEIGHT  POOL
lpar1    POWER8      shared    1     1    2   0.1  0.1    0.4    uncap     100  DefaultPool
$
```

For our example LPAR *lpar1*, the limits for the number of processor cores are at least 1 and at most 2. The current number of processor cores is 1. In the profile, 1 processor core is also configured as the desired value. The limits for the processor units are at least 0.1 and at most 0.4. The current value for the current processor units is 0.1. This is also configured in the profile as the desired value.

To dynamically change the memory, an active RMC connection is needed. This can be easily found out with the command "*lpar status*".

The number of processor cores should be increased by 1, the number of processor units by 0.1, dynamically as well as in the current profile. We quickly check if processor units are still available on the managed system. For this we need the associated managed system:

```
$ lpar show lpar1
LPAR      ID   SERIAL   TYPE      MS
lpar1     3    XXXXXXXX  aixlinux  ms01
$
```

The LPAR *lpar1* is located on the managed system *ms01*.

The managed system *ms01* still has enough unassigned processor units:

```
$ ms procs ms01
MS        INSTALLED  CONFIGURABLE  AVAILABLE
ms01      48.0       48.0          3.9
$
```

We now increase the processor configuration of *lpar1*:

```
$ lpar addproc lpar1 1
$ lpar addprocunits lpar1 0.1
$
```

The additional core and additional processor units are immediately available to the LPAR. The profile of the LPAR will automatically be changed accordingly, so that the same configuration will be used at the next activation with the current profile.

If the processor cores and processor units are only to be temporarily increased, the option "*-d*" (*dynamic only*) can be used. The old value is retained in the profile and the extension is only carried out dynamically:

```
$ lpar -d addproc lpar1 1
$ lpar -d addprocunits lpar1 0.1
$
```

If the LPAR is switched off and reactivated later, it will start again with 2 cores and 0.2 processor units.

If you only want to change the processor cores in the profile, the option "*-p <profile>*" is used instead of the option "*-d*":

```
$ lpar -p standard addproc lpar1 1
$ lpar -p standard addprocunits lpar 0.1
$
```

If processor cores and/or processor units are to be reduced, then the same applies to the command "*lpar rmproc*" or "*lpar rmprocunits*" with the same options as "*lpar addproc*" and "*lpar addprocunits*".

4. Changing Processor Limits in the Profile

The minimum and maximum number of cores and processor units of an LPAR can only be changed in the profile. For the change to take effect, the LPAR must be stopped and then restarted using the changed profile.

If the number of cores of *lpar1* is to be increased to 5, this is not possible at runtime due to the limit of the maximum number of cores of 4. In order to achieve the increase, the profile must be adjusted, and the limit increased to a value of at least 5. Only then can the desired number of cores of 5 be configured. A change of the processor limits in the profile can be carried out with the command "*lpar chproc*" for cores or "*lpar chprocunits*" for processor units:

```
$ lpar -p standard chproc lpar1 1 5 8
$ lpar -p standard chprocunits lpar1 0.1 0.5 2.0
$
```

In the example, the minimum value of the cores was left at 1, the desired number of cores was set to 5 and the maximum value was set to 8. For the processor units, the minimum value was 0.1, the maximum value 2.0 and the desired value 0.5. This leaves room for further increases later, which can then be performed dynamically up to a size of 8 cores and 2.0 processor units.

The LPAR must be reactivated with the changed profile after shutting down and powering off:

```
$ lpar -p standard activate lpar1
$
```

5. Configuring Physical Slots

Even physical slots can be configured in and out of an LPAR using DLPAR operations on a running LPAR.

Which physical slots are currently assigned to an LPAR can be displayed with the command "*lpar slots*":

```
$ lpar slots ms01-vio1
SLOT                      DRCIDX    DESCRIPTION
UXXX.001.XXXXXXX-P1-C12   21010019  PCIe2 16Gb 2-Port Fibre Channel Adapter
UXXX.001.XXXXXXX-P3-R1    21010021  RAID Controller
UXXX.001.XXXXXXX-P1-C4    21010038  Quad 8 Gigabit Fibre Channel LP Adapter
$
```

Of course, the configuration in the profile may differ, but it can also be easily viewed with the option "*-p <profile>*":

```
$ lpar -p standard slots ms01-vio1
SLOT                      DRCIDX    REQ  DESCRIPTION
UXXX.001.XXXXXXX-P1-C12   21010019  yes  PCIe2 16Gb 2-Port Fibre Channel Adapter
UXXX.001.XXXXXXX-P3-R1    21010021  yes  RAID Controller
UXXX.001.XXXXXXX-P1-C4    21010038  yes  Quad 8 Gigabit Fibre Channel LP Adapter
$
```

If an additional slot is to be configured into the LPAR, it is recommended to first list the available physical slots of the managed system:

```
$ ms slots ms01
SLOT                      DRCIDX    LPAR       DESCRIPTION
```

```
UXXX.001.XXXXXXX-P1-C10   21010010  -          PCIe2 16Gb 2-Port Fibre Channel Adapter
UXXX.001.XXXXXXX-P3-R2    21010011  ms01-vio2  RAID Controller
UXXX.001.XXXXXXX-P1-C12   21010019  ms01-vio1  PCIe2 16Gb 2-Port Fibre Channel Adapter
UXXX.001.XXXXXXX-P3-R1    21010021  ms01-vio1  RAID Controller
UXXX.001.XXXXXXX-P1-C4    21010038  ms01-vio1  Quad 8 Gigabit Fibre Channel LP Adapter
UXXX.001.XXXXXXX-P1-C2    21010048  ms01-vio2  Quad 8 Gigabit Fibre Channel LP Adapter
…
$
```

The output shows all physical slots of the managed system, as well as the current assignment to LPARs. To assign one of these slots, it must not currently be assigned to any other LPAR. Technically, when adding, the DRC index of the slot must be specified, which is shown in the output of the command "*ms slots*" above in a separate column. The assignment can be made using the DRC index and the command "*lpar addslot*":

```
$ lpar addslot lpar1 2101000A
$
```

The change is again carried out dynamically and in the current profile. The new hardware must then be configured into the operating system, which can be done by starting the *cfgmgr* under AIX as root:

```
lpar1 # cfgmgr
lpar1 #
```

If the slot is to be added dynamically only, the option "*-d*" can be used again:

```
$ lpar -d addslot lpar1 2101000A
$
```

Changing only the profile can be achieved with the option "*-p <profile>*":

```
$ lpar -p standard addslot lpar1 2101000A
$
```

The removal of physical slots works analogously with the command "*lpar rmslot*". It should be noted here, however, that all devices of the slot in the operating system must be removed (*rmdev*), before trying to remove the slot!

6. Virtual Ethernet Slots

Virtual Ethernet slots can be added to regular LPARs or virtual I/O servers. At first, we only consider normal LPARs, for which no so-called trunking adapters can be created.

To create a virtual Ethernet adapter you first need information about the available VSwitches and VLANs on the system. This information can be obtained with the command "*ms lsvswitch*":

```
$ ms lsvswitch ms01
MS   VSWITCH              SWITCH_MODE  VLAN_IDS
ms01  ETHPROD             VEB          720,735,437
ms01  ETHERNET0(Default)  VEB          100,102,105,107
ms01  ETHMGMT             VEB          1400,1600
```

```
$
```

First, we create a virtual Ethernet adapter for the VLAN 100 (default switch *ETHERNET0*). As a slot number on the LPAR we choose the slot 5 (this must not be used yet):

```
$ lpar addeth lpar1 5 100
$
```

Of course, this works only if there is already an Ethernet adapter in the LPAR with active RMC connection. Otherwise, the only option is to change the profile and reactivate the LPAR.

We create another virtual Ethernet adapter, this time in the VLAN 720 (VSwitch *ETHPROD*). Since the adapter does not belong to the default VSwitch this time, the VSwitch must be specified explicitly:

```
$ lpar addeth -s ETHPROD lpar1 6 720
$
```

Both adapters were dynamically configured (DLPAR operation) and the adapters were added to the current profile. The VLAN ID specified after the slot number is a so-called PVID (Port VLAN ID), which means that untagged Ethernet packets are tagged with the specified PVID. If you want to use more VLANs on the same adapter, then this is also possible. For this the option "*-i*" (IEEE) must be used, then up to 20 further VLANs are possible on the adapter:

```
$ lpar addeth -i lpar1 7 102 105,107
$
```

The additional VLANs are specified as a comma-separated list.

The result can be displayed again with the command "*lpar vslots*". Once for the active configuration:

```
$ lpar vslots lpar1
SLOT  REQ  TYPE    DATA
…
2     yes  eth     PVID=1400 VLANS= ETHMGMT
5     no   eth     PVID=100 VLANS= ETHERNET0
6     no   eth     PVID=720 VLANS= ETHPROD
7     no   eth     PVID=102 VLANS=105,107 ETHERNET0
…
$
```

and for the current profile:

```
$ lpar -p standard slots lpar1
SLOT  REQ  TYPE   DATA
…
2     yes  eth    PVID=1400 VLANS= ETHMGMT
5     no   eth    PVID=100 VLANS= ETHERNET0
6     no   eth    PVID=720 VLANS= ETHPROD
7     no   eth    PVID=102 VLANS=105,107 ETHERNET0
…
```

```
$
```

As before, with the option "*-d*" adapters can be configured dynamically only, and with the option "*-p <profile>*" into the profile only!

If you want to remove virtual Ethernet adapters again, there is the command "l*par rmeth*":

```
$ lpar rmeth lpar1 7
$
```

In addition to the LPAR, only the slot number must be specified.

7. Virtual SCSI Adapters

For virtual SCSI adapters, a virtual SCSI client adapter must be created in the client LPAR. In addition, a virtual SCSI server adapter must be created for this adapter on a virtual I/O server of the managed system. By default, the LPAR tool creates both the client and server adapters. I/Os in the client LPAR are then forwarded by the virtual SCSI client adapter, with the help of the hypervisor, to the virtual SCSI server adapter of a virtual I/O server, which then acts as a kind of storage system.

A virtual SCSI client adapter can be created with the command "*lpar addscsi*":

```
$ lpar addscsi lpar1 11 ms01-vio1 56
$
```

Where *11* is the slot number for the VSCSI client adapter on the client LPAR *lpar1* and *56* is the slot number to use for the VSCSI server adapter on the virtual I/O server *ms01-vio1*. The command "*lpar addscsi*" creates both the client adapter and the server adapter. In addition, client and server adapters are also entered in the respective profiles.

Let's take a quick look at the virtual adapters of the LPAR *lpar1*:

```
$ lpar vslots lpar1
SLOT  REQ  TYPE           DATA
0     yes  serial/server  remote: (any)/any hmc=1
1     yes  serial/server  remote: (any)/any hmc=1
5     no   eth            PVID=1234 VLANS= ETHERNET0
10    no   fc/client      remote: ms01-vio1(1)/55 c050760XXXXX0008,c050760XXXXX0009
11    no   scsi/client    remote: ms01-vio1(1)/56
$
```

Of course, each slot can only be used once! We'll take a look at what happens when trying to create a second VSCSI client adapter in slot *11*:

```
$ lpar addscsi lpar1 11 ms01-vio1 57
ERROR: lparAddScsiDynamic(): remote HMC command returned an error (1)
CMD on hmc01: chhwres -m ms01 -r virtualio –rsubtype scsi -o a -p lpar1 -s 11 -a
adapter_type=client,remote_lpar_name=ms01-vio1,remote_slot_num=57
StdErr: HSCL294C Dynamic add of virtual I/O resource failed:
StdErr: A Virtual I/O device already exists at slot 11.
```

```
$
```

The error message clearly states that there is already a device in slot *11*.

We use the unused slot *12* on the client side, but the slot *56* is already used on the virtual I/O server:

```
$ lpar addscsi lpar1 12 ms01-vio1 56
ERROR: lparAddScsiDynamic(): remote HMC command returned an error (1)
CMD on hmc01: chewers -m ms01 -r virtual –rsubtype scsi -o a -p ms01-vio1 -s 56 -a
adapter_type=server,remote_lpar_name=lpar1,remote_slot_num=12
StdErr: HSCL294C Dynamic add of virtual I/O resource failed:
StdErr: A Virtual I/O device already exists at slot 56.
$
```

Again, you get an error message. This time, the already used slot *56* on the virtual I/O server is reminded. However, the client adapter has already been created, as the output of "*lpar vslots*" shows:

```
$ lpar vslots lpar1
SLOT  REQ  TYPE           DATA
0     yes  serial/server  remote: (any)/any hmc=1
1     yes  serial/server  remote: (any)/any hmc=1
5     no   eth            PVID=1234 VLANS= ETHERNET0
10    no   fc/client      remote: ms01-vio1(1)/55 c050760XXXXX0008,c050760XXXXX0009
11    no   scsi/client    remote: ms01-vio1(1)/56
12    no   scsi/client    remote: ms01-vio1(1)/56
$
```

Before we delete the VSCSI client adapter in slot *12* of the client LPAR *lpar1*, we briefly look at the virtual I/O server *ms01-vio1*:

```
$ lpar vslots ms01-vio1 | grep lpar1
55    no   fc/server      remote: lpar1(10)/10
56    no   scsi/server    remote: lpar1(10)/11
$
```

The VSCSI server adapter in slot *56* is still connected to slot *11* of the client LPAR *lpar1*.

As with the creation of VSCSI adapters, the associated VSCSI server adapters on the virtual I/O server are also removed when deleting them.

We now delete the VSCSI client adapter in slot *12* of the client LPAR *lpar1*:

```
$ lpar rmscsi lpar1 12
INFO: remote virtual SCSI adapter is for different client adapter
INFO: remote virtual SCSI adapter is for different client adapter
$
```

The LPAR tool recognizes that the VSCSI server adapter in slot *56* of the virtual I/O server belongs to a different client adapter and therefore does not remove it, instead, a corresponding hint is issued. (This appears twice, because the adapter must be removed once dynamically and once from the profile.)

It is not mandatory to specify slot numbers when creating VSCSI adapters. You can also leave the selection of slot numbers to the LPAR tool. The LPAR tool then first determines free slot numbers and then uses these for creating the VSCSI adapters.

First we delete the VSCSI client adapter in slot *11* (including VSCSI server adapter):

```
$ lpar rmscsi lpar1 11
$
```

Now we create a VSCSI client adapter again, but this time we do not specify the slot number on the virtual I/O server:

```
$ lpar addscsi lpar1 11 ms01-vio1
lpar1 11 ms01-vio1 48
$
```

The slot number *48* was selected on the virtual I/O server! We don't show the output of "*lpar vslots lpar1*" and "*lpar vslots ms01-vio1*" here.

The selection of the slot number can be influenced by the attribute *LowestVirtualServerSlot* in one of the two configuration files */opt/pwrcmps/etc/lpar.cfg* or *~/.lpar.cfg*:

```
$ cat ~/.lpar.cfg
…
#LowestVirtualClientSlot 10

#LowestVirtualServerSlot 20
…
$
```

The configured value specifies, from which slot number the LPAR tool should start searching for a free slot on the virtual I/O server. The default is *20* here.

The slot number on the client can also be omitted and will be determined by the LPAR tool in a similar way. In this case, the selection can be controlled via the *LowestVirtualClientSlot* attribute.

It is possible to add a VSCSI client adapter only dynamically to an LPAR using the *-d* option:

```
$ lpar -d addscsi lpar1 12 ms01-vio1
lpar1 12 ms01-vio1 49
$
```

The slot number *49* for the VSCSI server adapter was again determined by the LPAR tool, but of course it could habe been specified as well.

The VSCSI client adapter was not added to the profile:

```
$ lpar -p standard vslots lpar1
SLOT  REQ  TYPE           DATA
0     1    serial/server  remote: -(any)/any status=unavailable hmc=1
```

```
1     1    serial/server  remote: -(any)/any status=unavailable hmc=1
5     0    eth            PVID=1234 VLANS= ETHERNET0
10    0    fc/client      remote: ms01-vio1(1)/55 c050760XXXXX0008,c050760XXXXX0009
11    0    scsi/client    remote: ms01-vio1(1)/48
$
```

On the virtual I/O server, the change is always carried out dynamically and in the currently active profile!

If a VSCSI client adapter is only to be configured in a profile, this can be done with the option "*-p <profile>*":

```
$ lpar -p standard -r addscsi lpar1 13 ms01-vio1 50
lpar1 13 ms01-vio1 50
$
```

In addition, we have specified the option, „*-r*“ for *required* here. The LPAR can then be activated with this profile only if the VSCSI client adapter in slot *13* is available.

As always, the associated VSCSI server adapter is created dynamically and in the profile on the virtual I / O server!

In rare cases, the automatic creation of VSCSI server adapters on a virtual I/O server is not desired (for example, because it already exists). With the option '*-c*' for *client-only*, the LPAR tool can be informed that only the client adapter should be created:

```
$ lpar -c addscsi lpar1 ms01-vio2
lpar1 14 ms01-vio2 26
$
```

(Slot *14* on the client and slot *26* on the virtual I/O server were selected by the LPAR tool, but the slot numbers can also be specified.)

No adapter was created on the virtual I/O server *ms01-vio2*:

```
$ lpar vslots ms01-vio2 | grep lpar1
$
```

The option, „-c“ can be combined with the options, „*-d*“ and, „*-p*“'.

Even when deleting a VSCSI client adapter, you can specify that only the client adapter should be deleted:

```
$ lpar -c rmscsi lpar1 14
$
```

8. Virtual FC Adapters

For virtual FC adapters, there is a client adapter in the LPAR and an associated server adapter on a virtual I/O server, similar to virtual SCSI. Again, the LPAR tool creates both the client and server adapters by default. The application is therefore analogous to the virtual SCSI adapters:

```
$ lpar addfc lpar1 20 ms01-vio2 55
$
```

Where *20* is the slot number for the virtual FC client adapter on the client LPAR *lpar1* and *55* is the slot number for the virtual FC server adapter on the virtual I/O server *ms01-vio2*. The command "*lpar addfc*" creates both the client adapter and the server adapter. In addition, client and server adapters are also entered in the respective profiles.

Let's take a quick look at the virtual adapters of the LPAR *lpar1*:

```
$ lpar vslots lpar1
SLOT  REQ  TYPE           DATA
0     yes  serial/server  remote: (any)/any hmc=1
1     yes  serial/server  remote: (any)/any hmc=1
5     no   eth            PVID=1234 VLANS= ETHERNET0
10    no   fc/client      remote: ms01-vio1(1)/55 c050760XXXXX0008,c050760XXXXX0009
11    no   scsi/client    remote: ms01-vio1(1)/56
20    no   fc/client      remote: ms01-vio2(2)/55 c050760XXXXX000e,c050760XXXXX000f
$
```

The virtual FC client adapter is automatically assigned 2 WWPNs, the first one is actively used as soon as a physical FC port is assigned to the associated virtual FC Server Adapter. The second WWPN is used for LPM (Live Partition Mobility).

An appropriate server adapter has been created on the virtual I/O server *ms-vio2*:

```
$ lpar vslots ms01-vio2 | grep lpar1
55  0  fc/server   remote: lpar1(10)/20
$
```

As with the VSCSI adapters, slot numbers do not have to be specified, but can be selected by the LPAR tool. We create another virtual FC client adapter and let the LPAR tool determine the slot number on the virtual I/O server:

```
$ lpar addfc lpar1 21 ms01-vio2
lpar1 21 ms01-vio2 26
$
$ lpar vslots lpar1
SLOT  REQ  TYPE           DATA
0     yes  serial/server  remote: (any)/any hmc=1
1     yes  serial/server  remote: (any)/any hmc=1
5     no   eth            PVID=1234 VLANS= ETHERNET0
10    no   fc/client      remote: ms01-vio1(1)/55 c050760XXXXX0008,c050760XXXXX0009
11    no   scsi/client    remote: ms01-vio1(1)/56
20    no   fc/client      remote: ms01-vio2(2)/55 c050760XXXXX000e,c050760XXXXX000f
21    no   fc/client      remote: ms01-vio2(2)/26 c050760XXXXX000a,c050760XXXXX000b
$
```

Here slot number *26* on the virtual I/O server *ms01-vio2* was selected by the LPAR tool. As with VSCSI, the selection can be controlled by the *LowestVirtualServerSlot* attribute in the configuration file.

The slot number on the client can also be determined by the LPAR tool.

All other options work the same way for virtual FC adapters as described for VSCSI adapters and will not be described here again.

The removal of a virtual FC adapter can be carried out with the command "*lpar rmfc*" analogous to the command "*lpar rmscsi*". We show here the removal of the adapter in slot *21*:

```
$ lpar rmfc lpar1 21
$
```

In addition to the client adapter, the server adapters on the virtual I / O server are also removed here.

If you accidentally deleted a client adapter, but you still need it, you can not just create a new adapter. The new adapter gets two new (not yet used) WWPNs. LUNs mapped to the old WWPNs will not be accessible with the new WWPNs (unless the storage configuration and SAN zoning are modified accordingly). However, it is possible to create a client adapter and specify the desired WWPNs. In the example we use the WWPNs *c050760XXXXX000a* and *c050760XXXXX000b* which were used by the adapter in slot 21 above:

```
$ lpar addfc lpar1 21 ms01-vio2 26 c050760XXXXX000a,c050760XXXXX000b
lpar1 21 ms01-vio2 26
$
```

The virtual FC adapter in slot *21* of the client LPAR *lpar1* has the specified WWPNs:

```
$ lpar vslots lpar1
SLOT  REQ  TYPE           DATA
0     yes  serial/server  remote: (any)/any hmc=1
1     yes  serial/server  remote: (any)/any hmc=1
5     no   eth            PVID=1234 VLANS= ETHERNET0
10    no   fc/client      remote: ms01-vio1(1)/55 c050760XXXXX0008,c050760XXXXX0009
11    no   scsi/client    remote: ms01-vio1(1)/56
20    no   fc/client      remote: ms01-vio2(2)/55 c050760XXXXX000e,c050760XXXXX000f
21    no   fc/client      remote: ms01-vio2(2)/26 c050760XXXXX000a,c050760XXXXX000b
$
```

This allows the old LUNs to be accessed again.

6. Virtual I/O Server

A large number of operations on a virtual I/O server can be conveniently performed via the LPAR tool, without having to log in directly to the virtual I/O server.

1. Virtual Media Repository

Here we show the administration of a virtual media repository on a virtual I/O server using the LPAR tool.

First, we create a repository with a size of 20 GB:

```
$ vios mkrep ms04-vio2 20g
Virtual Media Repository Created
Repository created within "VMLibrary" logical volume
$
```

Details about the repository can be viewed with the command "*vios lsrep*":

```
$ vios lsrep ms04-vio2
Size(mb) Free(mb) Parent Pool         Parent Size      Parent Free
   20396    20396 rootvg                  279552           172544
$
```

To use the Virtual Optical Library, an LPAR requires a virtual CD drive. A virtual CD drive is a virtual SCSI device, this means the LPAR requires a virtual SCSI adapter. Our example LPAR already has such an adapter:

```
$ lpar vslots lpar6
SLOT  REQ  TYPE           DATA
0     1    serial/server  remote: -(any)/any status=unavailable hmc=1
1     1    serial/server  remote: -(any)/any status=unavailable hmc=1
5     0    eth            PVID=1234 VLANS=- XXXXXXXXXXXX ETHERNET0
10    0    fc/client      remote: ms04-vio1(1)/23 c050760XXXXX0016,c050760XXXXX0017
20    0    fc/client      remote: ms04-vio2(2)/23 c050760XXXXX0018,c050760XXXXX0019
21    no   scsi/client    remote: ms04-vio2(3)/21
$
```

To create a virtual CD drive (file backed optical), we need to know the *vhost* adapter on the virtual I/O server for the VSCSI client adapter:

```
$ vios vscsi ms04-vio2
VIOS       SLOT   NAME        CLIENT               LUNS
ms04-vio2  C19    vhost0      0x00000006           0
ms04-vio2  C21    vhost1      0x00000001           0
$
```

The correct adapter is *vhost1* (slot *C21*). Now we can create the virtual CD drive:

```
$ vios mkfbo ms04-vio2 vhost1
vtopt0 Available
$
```

If desired, the virtual optical device can also be given a descriptive name:

```
$ vios mkfbo ms04-vio2 vhost1 lpar6_cd
lpar6_cd Available
$
```

The devices can be displayed with the command "*vios lsvopt*":

```
$ vios lsvopt ms04-vio2
VTD             Media                                           Size(mb)
lpar6_cd        No Media                                             n/a
vtopt0          No Media                                             n/a
$
```

We have created some media in the virtual media repository (not yet supported by the LPAR tool):

```
$ vios lsrep ms04-vio2
Size(mb) Free(mb) Parent Pool         Parent Size      Parent Free
   51703    45887 rootvg                  571392           286720

Name                                                File Size Optical         Access
aix610702-mksysb                                         1752 None            ro
aix710103-mksysb                                         1643 None            ro
aix710104-mksysb                                         1653 None            ro
blank                                                     768 None            rw
$
```

A virtual optical medium can be inserted into a virtual CD drive with the command "*vios loadopt*":

```
$ vios loadopt ms04-vio2 lpar6_cd blank
$
```

Which medium is currently inserted in a virtual drive can be seen as follows:

```
$ vios lsrep ms04-vio2
Size(mb) Free(mb) Parent Pool         Parent Size      Parent Free
    1015      915 rootvg                  139776            99072

Name                                                File Size Optical         Access
blank                                                    100 lpar6_cd         rw
```

The virtual medium is now available in the client LPAR.

To "*eject*" a medium again, "*vios unloadopt*" can be used:

```
$ vios unloadopt ms04-vio2 lpar6_cd
$
```

If the capacity of the repository is exhausted, it can be easily extended:

```
$ vios chrep ms04-vio2 10g
$
```

Of course, this is only possible while there is still room in the *rootvg*.

If the virtual media repository is to be deleted, this is also possible with the LPAR tool:

```
$ vios rmrep ms04-vio2
ERROR: viosRmrep(): remote HMC command returned an error (1)
CMD on hmc01: viosvrcmd -m ms04 -p ms04-vio2 -c "rmrep"
StdErr: HSCL2970 The IOServer command has failed because of the following reason:
StdErr: DVD repository contains file backed DVD media. Use -f to remove
StdErr:
StdErr: rc=4
$
```

However, the media must be deleted beforehand. Alternatively, you can also use the option "*-f*" to force deletion of all media when deleting the virtual media repository:

```
$ vios rmrep -f ms04-vio2
$
```

2. Administration of VSCSI

On the virtual I/O server a *vhost* device must be created for each virtual SCSI server adapter:

```
$ vios vscsi ms01-vio1
VIOS        SLOT  NAME    CLIENT  LUNS
ms01-vio1   C35   vhost0  0x04    18
ms01-vio1   C80   vhost1  0x05    7
ms01-vio1   C48   vhost2  0x03    0
$
```

Slot *48* belongs to the device *vhost2*. There are no virtual target devices assigned to this *vhost* device.

Suppose that the disk *hdisk15* is not yet in use on the virtual I/O server and should be assigned to LPAR *lpar1*, e.g. as a disk for the *rootvg*, then you can perform the assignment with the following command:

```
$ vios map ms01-vio1 vhost2 hdisk15
$
```

A quick check shows that a LUN is now associated with the *vhost2* device:

```
$ vios vscsi ms01-vio1
VIOS        SLOT  NAME    CLIENT  LUNS
ms01-vio1   C35   vhost0  0x04    18
ms01-vio1   C80   vhost1  0x05    7
ms01-vio1   C48   vhost2  0x03    1
$
```

More detailed information can be obtained by specifying the *vhost* device as an additional argument:

```
$ vios vscsi ms01-vio1 vhost2
VIOS        VHOST   DISK     VTD
ms01-vio1   vhost2  hdisk15  vtscsi2
$
```

The name *vtscsi2* is a bit uninformative, you can not see what it is used for. Many administrators give descriptive names indicating the purpose of the device in the client LPAR, e.g. *lpar1_hd0* to indicate that this device is used in the LPAR *lpar1* as *hdisk0*. Of course you can also specify any name for the virtual target device when mapping with the LPAR tool:

```
$ vios map ms01-vio1 vhost2 hdisk16 lpar1_hd1
$
```

Such a name is very helpful, especially if a lot of disks are used with VSCSI:

```
$ vios vscsi ms01-vio1 vhost2
VIOS        VHOST   DISK     VTD
ms01-vio1   vhost2  hdisk15  vtscsi2
ms01-vio1   vhost2  hdisk16  lpar1_hd1
$
```

Of course, such a mapping can also be removed again. For this, the corresponding *hdisk* in the client LPAR should first be unconfigured from the operating system. To delete the mapping on the virtual I/O server, use the command "*vios unmap*":

```
$ vios unmap ms01-vio1 vhost2 lpar1_hd1
$
```

3. Administration of VFC (NPIV)

On the virtual I/O server, a *vfchost* device is created for every virtual FC server adapter. We can list these devices using the command "*vios npiv*":

```
$ vios npiv ms01-vio2
VIOS        ADAPT  NAME      CLIENT  OS   ADAPT  STATUS        PORTS
ms01-vio2   fcs3   vfchost1  lpar2   AIX  fcs1   LOGGED_IN       7
ms01-vio2   fcs1   vfchost2  lpar3   AIX  fcs1   LOGGED_IN       5
ms01-vio2          vfchost3  lpar1   AIX  fcs1   NOT_LOGGED_IN   0
$
```

The *vfchost* device for spar 1 is *vfchost3*. The *vfchost3* device is not yet assigned to a physical adapter (column 2 *ADAPT*). Therefore, the status is *NOT_LOGGED_IN* (not logged into the fabric) and the number of ports is 0.

There is no assignment to a physical port. This can easily be done with the command "*vios vfcmap*":

```
$ vios vfcmap ms01-vio2 vfchost3 fcs3
$
```

The status and assignment of the adapter should have changed:

```
$ vios npiv ms01-vio2
VIOS       ADAPT  NAME      CLIENT  OS   ADAPT  STATUS       PORTS
ms01-vio2  fcs3   vfchost1  lpar2   AIX  fcs1   LOGGED_IN      7
ms01-vio2  fcs1   vfchost2  lpar3   AIX  fcs1   LOGGED_IN      5
ms01-vio2  fcs3   vfchost3  lpar1   AIX  fcs1   LOGGED_IN      1
$
```

The adapter is now in status *LOGGED_IN* and a port is displayed. The first WWPN of the virtual FC Client Adapter is now additionally mapped to the physical port *fcs3* via NPIV. This WWPN is logged into the fabric and is currently the only port in its zone.

Storage can now be zoned directly to the WWPN of the client LPAR. The disks no longer have to be laboriously mapped on the virtual I/O servers as in VSCSI.

In order to remove the mapping of the *vfchost* device to a physical port, the "*vios vfcmap*" command is executed without specifying a physical port:

```
$ vios vfcmap ms01-vio2 vfchost3
$
```

7. Administration of Managed Systems

1. Multiple Shared Processor Pools

Since the introduction of POWER6 servers there is the possibility to configure several shared processor pools. Thus, LPARs can be organized into shared processor pools, e.g. to have better control over the performance or to restrict software licenses to LPARs in a shared processor pool.

Which shared processor pools are available on a managed system can be listed with the command "*ms lsprocpool*":

```
$ ms lsprocpool ms09
MS          PROCPOOL                  MAX   RESERVED
ms09        DefaultPool(0)              -          -
ms09        pool1(1)                  4.0        0.0
ms09        pool2(2)                  1.0        0.0
$
```

(Note: several managed systems can be specified)

The ID of a processor pool is displayed in parentheses after the custom pool name. The processor pool *DefaultPool* with ID *0* always exists and is always active. The *MAX* column lists the maximum pool capacity, which is always an integer and indicates the maximum number of *procuncits* in the processor pool. The LPARs in a shared processor pool can not exceed this maximum capacity. The column *RESERVED* indicates how many *procunits* are reserved for LPARs in the shared processor pool (in addition to the entitled capacity of the LPARs).

The output can be defined by means of the options "*-f*" and/or "*-F*":

```
$ ms lsprocpool -f ms09
ms09:
   name=DefaultPool
   shared_proc_pool_id=0
   \"lpar_names=lpar2,lpar1,ms09-vio2,ms09-vio1,lpar3\"
   \"lpar_ids=4,3,2,1,7\"

ms09:
   name=pool1
   shared_proc_pool_id=1
   max_pool_proc_units=4.0
   curr_reserved_pool_proc_units=0.0
   pend_reserved_pool_proc_units=0.0
   lpar_ids=none

ms09:
   name=pool2
   shared_proc_pool_id=2
   max_pool_proc_units=1.0
   curr_reserved_pool_proc_units=0.0
   pend_reserved_pool_proc_units=0.0
   lpar_ids=none

$
```

```
$ ms lsprocpool -F name:max_pool_proc_units ms09
name:max_pool_proc_units
```

```
DefaultPool:-
pool1:4.0
pool2:1.0
$
```

By default, each managed system has up to 64 shared processor pools, but by default all processor pools are inactive except for the default pool. All shared processor pools with a maximum pool capacity of 0 are inactive. With the option "*-a*" all shared processor pools, including the inactive ones, can be displayed:

```
$ ms lsprocpool -a ms09
MS         PROCPOOL                  MAX   RESERVED
ms09       DefaultPool(0)              -          -
ms09       pool1(1)                  4.0        0.0
ms09       pool2(2)                  1.0        0.0
ms09       SharedPool03(3)           0.0        0.0
ms09       SharedPool04(4)           0.0        0.0
ms09       SharedPool05(5)           0.0        0.0
ms09       SharedPool06(6)           0.0        0.0
…
$
```

2. Administering Virtual Ethernet Switches

By default, every managed system always has the virtual ethernet switch *ETHERNET0*. Typically, LPARs are then created using virtual ethernet adapters attached to this VSwitch. The network traffic is then forwarded to a so-called trunking port on a virtual I/O server, which is part of a shared ethernet adapter (SEA). Each SEA has at least one physical Ethernet adapter by which the network traffic is then forwarded to a physical LAN.

If another shared ethernet adapter is now configured with a physically separate LAN, the separation of the networks should also be retained in the managed system. However, if one uses the default VSwitch *ETHERNET0*, which is already used for the previous SEA, then the externally separated networks are connected internally by the VSwitch *ETHERNET0*. This is not only for security-related aspects of concern, but can also lead to surprises when the same VLAN IDs are used in the physically separate networks.

The existing virtual ethernet switches on a managed system can be listed using "*ms lsvswitch*":

```
$ ms lsvswitch ms01
MS    VSWITCH              SWITCH_MODE  VLAN_IDS
ms01  ETHERNET0(Default)   VEB          100,200,300
$
```

For each VSwitch, the available (configured) VLANs are also listed.

To create a new VSwitch, the command "*ms addvswitch*" is used, as an argument the name of the managed system and the name of the new VSwitch have to be specified:

```
$ ms addvswitch ms01 ETHNAS
$
```

The new VSwitch is called *ETHNAS* and will of course be listed from now on:

```
$ ms lsvswitch ms01
MS    VSWITCH              SWITCH_MODE  VLAN_IDS
ms01  ETHERNET0(Default)   VEB          100,200,300
ms01  ETHNAS               VEB          none
$
```

By default, a new VSwitch is created in *VEB* (Virtual Ethernet Bridge) mode. The VSwitch then works like an Ethernet bridge, LPARs in the same managed system can communicate directly via the VSwitch, without the network packets going through an SEA into the external physical network. This means that network traffic between LPARs in the same managed system using the same VSwitch and the same VLAN are not inspected by a firewall!

In the *VEPA* (Virtual Ethernet Port Aggregator) mode, all network traffic is forwarded to the external network. If the destination is in the same managed system as the sender, the network packets are routed back to the managed system. LPARs in the same managed system can no longer exchange network packets directly via the hypervisor. Of course, the performance is worse than in the *VEB* mode.

When creating virtual ethernet adapters, the option '*-s*' can be used to specify the desired VSwitch.

If a VSwitch is no longer needed, it can be removed again using "*ms rmvswitch*":

```
$ ms rmvswitch ms01 ETHNAS
$
```

However, the VSwitch may not be in use, otherwise you will get the following error message:

```
$ ms rmvswitch ms01 ETHNAS
ERROR: msVswitch(): remote HMC command returned an error (1)
CMD on hmc01: chhwres -m ms01 -r virtualio –rsubtype vswitch -o r –vswitch ETHNAS
StdErr: HSCL3689 This virtual switch cannot be deleted since the following virtual networks
are using this virtual switch: VLAN100-ETHNAS,. All virtual networks using a virtual switch
must be deleted before the virtual switch can be deleted.
$
```

3. Managing Partition Data

The partition profile data can be saved on the HMC. If an LPAR has been deleted or the configuration of an LPAR has been changed, the LPAR can be restored by means of a backup.

To save the profile data of all LPARs of a managed system on an HMC, use the command "*ms bkprofdata*":

```
$ ms bkprofdata ms01 backup01
$
```

As arguments, the managed system and a file name for the backup must be specified. If a relative path is specified, the backup is stored on the HMC under */var/hsc/profiles/<serial-number>*, where *<serial-number>* is the serial number of the managed system.

Caution: The profile data of all LPARs are always backed up, the backup can not be restricted to individual LPARs!

Which backups of profile data are already available in the named default directory can be listed with "*ms lsprofdata*":

```
$ ms lsprofdata ms01
backup
backup01
$
```

On the managed system *ms01* there are currently the following LPARs, which are all included in the above backup:

```
$ lpar -m ms01 show
LPAR             ID  SERIAL    LPAR_ENV   MS    HMCs
lpar2            4   XXXXXXX4  aixlinux   ms01  hmc01,hmc02
lpar3            5   XXXXXXX5  aixlinux   ms01  hmc01,hmc02
ms01-vio1        2   XXXXXXX2  vioserver  ms01  hmc01,hmc02
ms01-vio2        3   XXXXXXX3  vioserver  ms01  hmc01,hmc02
$
```

The LPAR *lpar3* is not needed and is currently not active. We delete these to demonstrate a restore of the profile data:

```
$ lpar delete lpar3
Deleting LPAR lpar3
done
ms01-vio1
Remove virtual fc server adapters on ms01-vio1
done
ms01-vio2
Remove virtual fc server adapters on ms01-vio2
done
Remove virtual fc server adapters on ms01-vio2
Done
$ lpar -m ms01 show
LPAR             ID  SERIAL    LPAR_ENV   MS          HMCs
lpar2            4   XXXXXXX4  aixlinux   ms01  hmc01,hmc02
ms01-vio1        2   XXXXXXX2  vioserver  ms01  hmc01,hmc02
ms01-vio2        3   XXXXXXX3  vioserver  ms01  hmc01,hmc02
$
```

Now the created backup will be used to restore the LPAR *lpar3*:

```
$ ms rstprofdata ms01 3 backup01
$
```

There are the following 4 restore types:

1 — full restore

2 — Merging the current configuration with the backup, in case of differences the backup will be used

3 — Merging the current configuration with the backup, in case of differences the current configuration is maintained

4 — Initialization, all partitions, partition profiles and system profiles are deleted.

The LPAR *lpar3* has been restored. In order that it will be included again in the mapping files, once the command "*hmc rescan*" has to be started:

```
$ hmc rescan
ms01
ms02
ms03
ms04
…
$
```

The LPAR *lpar3* has been recreated:

```
$ lpar -m ms01 show
LPAR             ID  SERIAL    LPAR_ENV   MS    HMCs
lpar2            4   XXXXXXX4  aixlinux   ms01  hmc01,hmc02
lpar3            5   XXXXXXX5  aixlinux   ms01  hmc01,hmc02
ms01-vio1        2   XXXXXXX2  vioserver  ms01  hmc01,hmc02
ms01-vio2        3   XXXXXXX3  vioserver  ms01  hmc01,hmc02
$
```

If a backup is no longer needed, it can be deleted again using "*ms rmprofdata*":

```
$ ms rmprofdata ms01 backup01
$
```

Caution: A restore may affect all LPARs, although not shown in the example above. A backup usually always contains the profile data of several LPARs.

8. HMC

All HMC user accounts can be managed using the LPAR tool. Each HMC user has assigned a resource role and a task role. The resource role determines which managed systems and LPARs the user can manage. Managed systems and LPARs that are not assigned to their resource role are not visible to them. This is true for the command line (and thus the LPAR tool) as well as for the GUI. The task role determines which operations the user is allowed to perform on managed systems, LPARs and HMCs.

1. User Accounts

First, we'll show how to use the LPAR tool to list, create, delete, and modify user accounts.

A list of currently created users on an HMC can be obtained with the command "*hmc lshmcusr*":

```
$ hmc lshmcusr hmc01
NAME      DESCRIPTION     TASKROLE        RESOURCEROLE
hscroot   HMC Super User  hmcsuperadmin   ALL:
lpar2rrd  technical user  hmcviewer       ALL:
operator  Operators       firstlevel      ALL:
kmeier    Klaus Meier     hmcsuperadmin   ALL:
…
$
```

The resource role "*ALL:*" includes all managed systems and LPARs.

A new user can be created with the command "*hmc mkhmcusr*":

```
$ hmc mkhmcusr hmc01 testuser
Enter the new password: XXXXXXXX
Retype the new password: XXXXXXXX
$ hmc lshmcusr hmc01
NAME      DESCRIPTION     TASKROLE        RESOURCEROLE
hscroot   HMC Super User  hmcsuperadmin   ALL:
lpar2rrd  technical user  hmcviewer       ALL:
operator  Operators       firstlevel      ALL:
kmeier    Klaus Meier     hmcsuperadmin   ALL:
…
testuser                  hmcviewer       ALL:
$
```

The new user is assigned the task role "*hmcviewer*" and the resource role "*ALL:*".

Of course, the attributes of a user, such as assigned task role or resource role, can also be changed easily with the LPAR tool. For this there is the command "*hmc chhmcusr*":

```
$ hmc chhmcusr hmc01 testuser taskrole=hmcsuperadmin
$ hmc lshmcusr hmc01
NAME      DESCRIPTION     TASKROLE        RESOURCEROLE
hscroot   HMC Super User  hmcsuperadmin   ALL:
lpar2rrd  technical user  hmcviewer       ALL:
operator  Operators       firstlevel      ALL:
kmeier    Klaus Meier     hmcsuperadmin   ALL:
…
testuser                  hmcsuperadmin   ALL:
```

```
$
```

It is just as easy to change the resource role of a user:

```
$ hmc chhmcusr hmc01 testuser resourcerole=testonly
$ hmc lshmcusr hmc01
NAME      DESCRIPTION     TASKROLE        RESOURCEROLE
hscroot   HMC Super User  hmcsuperadmin   ALL:
lpar2rrd  technical user  hmcviewer       ALL:
operator  Operators       firstlevel      ALL:
kmeier    Klaus Meier     hmcsuperadmin   ALL:
…
testuser                  hmcsuperadmin   testonly
$
```

Of course, the resource role "*testonly*" must exist!

If a user account is no longer needed, it can be deleted:

```
$ hmc rmhmcusr hmc01 testuser
$
```

2. Resource Roles

Resource roles can be used to restrict the managed systems and LPARs a user is allowed to administer. Managed systems and LPARs that are not part of the user's resource role are not visible to the user. These are displayed neither in the GUI nor on the command line.

By default, only the built-in and unchangeable resource role "*ALL:*" exists. This can be used, but is not visible to many commands. Additional resource roles can be created, configured and assigned to users as desired.

A list of existing resource roles on an HMC can be obtained with the command "*hmc lsresourceroles*":

```
$ hmc lsresourceroles hmc01
NAME
role01
role02
ms01only
testonly
$
```

The built-in resource role "*ALL:*" is not indicated as already indicated above. Only resource roles that can be changed are displayed.

If you want to see the resources inside a resource role, use the command "*hmc lsresourcerole*". The resource role to be displayed is simply given as an argument after the HMC name:

```
$ hmc lsresourcerole hmc01 testonly
RESOURCEROLE: testonly
cec:ms01
lpar:all:ms01
lpar:testdb01
lpar:testdb02
$
```

To add a new resource role, the command "*hmc mkresourcerole*" can be used:

```
$ hmc mkresourcerole hmc01 myrole
$
```

The resource role is empty after creation:

```
$ hmc lsresourcerole hmc01 myrole
RESOURCEROLE: myrole
$
```

Changing an existing resource role can be done with the command "*hmc chresourcerole*". There are a number of different options.

Adding a managed system:

```
$ hmc chresourcerole hmc01 myrole +cec:ms02
$
```

If the name of the managed system is unique, i.e. is not at the same time also the name of an LPAR, then the short form:

```
$ hmc chresourcerole hmc01 myrole +ms02
$
```

can be used.

Adding an LPAR works similarly:

```
$ hmc chresourcerole hmc01 myrole +lpar:lpar1
$
 oder kürzer
$ hmc chresourcerole hmc01 myrole +lpar1
$
```

If all LPARs of a managed system are addressed, this is done with:

```
$ hmc chresourcerole hmc01 myrole +lpar:all:ms02
$
```

Of course, by prefixing a minus sign, you can remove LPARs or managed systems from a resource role:

```
$ hmc chresourcerole hmc01 testrole -ms04
$ hmc chresourcerole hmc01 testrole -cec:ms04
$ hmc chresourcerole hmc01 testrole -lpar:lpar3
$ hmc chresourcerole hmc01 testrole -lpar3
$ hmc chresourcerole hmc01 testrole -lpar:all:ms03
```

Resource roles that are no longer needed can be easily deleted using "*hmc rmresourcerole*":

```
$ hmc rmresourcerole hmc01 testrole
$
```

3. Task Roles

Task roles allow you to configure which operations a user may perform on managed systems, LPARs or HMCs.

By default, some task roles are already predefined on each HMC. Additional task roles can be created. The currently existing Task Roles can be listed with the command "*hmc lstaskroles*":

```
$ hmc lstaskroles hmc01
NAME           PARENT
hmcdev         Predefined
hmcsuperadmin  Predefined
hmcviewer      Predefined
$
```

Each task role can be listed in detail:

```
$ hmc lstaskrole hmc01 hmcviewer
taskrole: hmcviewer
parent: Predefined
resources:
  frame
    CheckPSN
    ListFrameProperty
  cec
    CaptureSystemTemplate
    CollectCECVPDInfo
    LSProfileSpace
    ListCECProperty
    ListCoDInformation
    ListPCIeTopology
    ListRioTopology
    ListSSP
    ListSystemProfileProperty
    ListUtilizationData
    ListVETInfo
    ManageSysProfile
    ViewDumps
    ViewPowerManagment
    ViewSSP
  lpar
    CapturePartitonTemplate
    ListLPARProperty
    ListProfileProperty
    ManageProfile
  HMCConsole
    ChangeLocale
    ChangeUserPasswords
    CollectVPDInfo
    ListCloudConnServiceSettings
```

```
    ListConnections
    ListHMCConfiguration
    ListHMCEncrTask
    ListSNMPServiceableEvents
    ListStorageMedia
    TemplateLibrary
    TipOfTheDay
    UserSettings
    ViewConsoleEvents
    ViewHMCFileSystems
$
```

Each task role has a parent. The resources of the parent task role limit which resources can be assigned to a resource role. This prevents an escalation of privileges.

A new task role can be created with the command "*hmc mktaskrole*":

```
$ hmc mktaskrole hmc01 trole1 hmcviewer
$ hmc lstaskrole hmc01 trole1
taskrole: trole1
parent: hmcviewer
resources:
$
```

The new task role is initially empty. Only resources (operations) can be added to the new task role that are included in the parent role *hmcviewer*.

Resources can be added to a task role with the command "*hmc chtaskrole*":

```
$ hmc chtaskrole hmc01 trole1 +lpar:ListLPARProperty
$ hmc chtaskrole hmc01 trole1 +cec:ListCECProperty
$ hmc lstaskrole hmc01 trole1
taskrole: trole1
parent: hmcviewer
resources:
  cec
    ListCECProperty
  lpar
    ListLPARProperty
$
```

Of course you can also remove resources from a task role:

```
$ hmc chtaskrole hmc01 trole1 -cec:ListCECProperty
$ hmc chtaskrole hmc01 trole1 -lpar:ListLPARProperty
$
```

Unnecessary task roles can be deleted with the help of the command "*hmc rmtaskrole*":

```
$ hmc rmtaskrole hmc01 trole1
$
```

4. Users logged into the HMC

Users can either work on the HMC using the GUI or the CLI. Actions taken by logged in users may occasionally hang. This sometimes happens especially when working with the GUI.

Users logged in via CLI can be displayed with the command "*hmc lslogon*", including the so-called tasks that started them:

```
$ hmc lslogon hmc01
USER_NAME  TTY_ID  LOGON_TIME        ACCESS_LOCATION
   TASK_NAME TTY_ID  START_TIME          USER_NAME   PID
kmeier     pts/1   2018-05-29 16:17  10.11.12.13
   bash      pts/1   May 29 16:17:25 2018 root        20513
$
```

Should a process hang or the logged-in user is unavailable for a longer time, the process can be terminated:

```
$ hmc termtask hmc01 20513
$
```

The specified argument is the PID of the process to be stopped.

The users logged in on the GUI can also be listed with "*hmc lslogon*", the option "*-r webui*" must be specified in that case:

```
$ hmc lslogon -r webbi hmc01
USER_NAME  SESSION_ID  LOGON_TIME           LOGON_MODE
   TASK_ID  TASK_NAME  SESSION_ID  START_TIME           USER_NAME
kmeier     3           04/26/2018 12:54:28  Enhanced+
   156      ms01       3           04/26/2018 12:59:13  kmeier
$
```

The default for the *-r* option, if not specified, is "*-r ssh*".

You can also terminate a task on the GUI with the "*hmc termtask*" command. To do this, the displayed task ID must be specified as an argument:

```
$ hmc termtask -r webui hmc01 156
$
```

9. Troubleshooting

If error messages occur while using the LPAR tool, then of course the question arises as to the cause of the error. In general, errors can be assigned to one of the following classes:

A command was not used correctly: e.g. wrong spelling, wrong syntax, wrong option, missing arguments or similar. In this case, an error message should be output. The command should be corrected and then restarted.

A command was used correctly, but the executing HMC returns an error.

The LPAR tool does not work properly and returns an error message.

All calls to the LPAR tool are logged in the file *~/lpar.log*, in case of errors also the error message is recorded in the log file.

1. Incorrect Usage of Commands

If a command is not used correctly, the LPAR tool issues an error message together with a usage message.

```
$ lpar vslots
ERROR: argument missing
   Usage: lpar [option …] keyword [arg …]
          lpar -V
…
$
```

The LPAR tool issues an error message:

„*ERROR: argument missing*“.

Often an LPAR name or managed system name is misspelled:

```
$ lpar addprocs testlpar7 1
ERROR: LPAR ‚testlpar‘ not found
$
```

2. HMC returns Error Message

Even if commands are used correctly, errors can occur.

```
$ ms sriov ms03
ERROR: remote HMC command returned an error (1)
CMD on hmc01: lshwres -m ms03 -r sriov –rsubtype adapter
StdErr: HSCL1237 The managed system does not support SR-IOV.
$
```

The ms command was used correctly, but the specified managed system *ms03* does not support *SRIOV*. The LPAR tool gives the error message

„*ERROR: remote HMC command returned an error (1)*“

On the associated HMC (*hmc01*) the command

```
lshwres -m ms03 -r sriov –rsubtype adapter
```

was executed with the correct syntax. The LPAR tool outputs the error message of the HMC command:

„*StdErr: HSCL1237 The managed system does not support SR-IOV.*“

There is no error in the LPAR tool, the specified managed system does not support *SRIOV*.

Here is an example of an error in the LPAR tool (this is however already fixed in the current version):

```
$ lpar vslots lpar7
ERROR: remote HMC command returned an error(1)
CMD on hmc02: lshwres -m ms11 -r virtualio –rsubtype vnic –level lpar –filter
lpar_names=lpar7
StdErr: The command entered is either missing a required parameter or a parameter value is
invalid. The parameters that are missing or have an invalid value are –rsubtype. Please
check your entry and retry the command.
$
```

Again, the command was used correctly (command *lpar*). However, in this case the remote command on the HMC is faulty and thus the virtual slots of the LPAR *lpar7* can not be displayed. The problem in this case is an older HMC version that does not support *vNICs* and therefore does not know the value "*vnic*" for the option "*-rsubtype*". The problem was resolved by installing a check.

So if the LPAR tool generates an HMC command that is not correct (invalid option or argument), then there is probably an error in the program code of the LPAR tool. The problem can then be reported to *support@powercampus.de*. The used HMC version as well as the output of the command including the error message should then be sent.

3. Errors of the LPAR tool

No software is perfect and error-free. We have subjected the LPAR tool to intensive testing, but it can not be ruled out that there are errors in the LPAR tool. Here is a constructed error to demonstrate how such an unknown error could show up:

```
$ lpar lsrefcode lpar5
ERROR: tableGetRow(): index out of range
$
```

To avoid crashes of the LPAR-Tool commands, most of the functions used check the arguments with which they were called. If a faulty argument is detected, the command aborts with an error message and an indication of the function that detected the error:

„*ERROR: tableGetRow(): index out of range*“

Here, a function called *tableGetRow()* was called, when checking its arguments, it found the given index was out of range. Error messages that include a function name usually indicate an error in one of the commands of the LPAR tool. Also in this case, the error should be reported to *support@powercampus.de* along with the screen output.

A. Commandreference - hmc

All operations directly related to an HMC are performed by the LPAR tool with the *hmc* command. For most commands, the first argument after the options and the keyword must be the name of an HMC.

To view the version of the LPAR tool:

```
$ hmc -V
Version: 1.2.0
(c) 2017-2018 by PowerCampus
$
```

1. hmc add

Registering (adding) a new HMC:

```
hmc add [-u <user>] [-v] <hmc>
  -u <user> # der User-Account auf der HMC
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

If the option "*-u <user>*" is not used then the current username will be used.

2. hmc chhmcusr

Changing a user account on an HMC:

```
hmc chhmcusr [-v] <hmc> <user> <attribute>=<value> …
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

3. hmc chresourcerole

Changing a resource role on an HMC:

```
hmc chresourcerole [-v] <hmc> <resourcerole> {+|-}<resource>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

4. hmc chtaskrole

Changing a task role on an HMC:

```
hmc chtaskrole [-v] <hmc> <taskrole> {+|-}<resource> …
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

5. hmc connections

Listing of all established SSH-master connections to the HMCs:

```
hmc connections
```

6. hmc disconnect

Diconnecting a SSH-master connection to an HMC:

```
hmc disconnect <hmc>
```

7. hmc help

Overview of the available help:

```
hmc help [<keyword>|user|usage]
```

8. hmc list

Listing of registered HMCs:

```
hmc list
```

The command does not execute a remote command on an HMC, but accesses a local mapping file containing the information.

9. hmc lsfirewall

Show the firewall configuration of an HMC:

```
hmc lsfirewall [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

10. hmc lshmcfs

Show the usage of HMC filesystems:

```
hmc lshmcfs [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

11. hmc lshmcusr

Show all users on an HMC:

```
hmc lshmcusr [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

12. hmc lslic

List all system firmware files available on the HMC:

```
hmc lslic [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

13. hmc lslogon

View logged-in users and their tasks on an HMC:

```
hmc lslogon [-r ssh|webui] [-v] <hmc>
  -r ssh # per SSH eingeloggte Benutzer und Tasks anzeigen (default)
  -r webui # per GUI eingeloggte Benutzer und Tasks anzeigen
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

14. hmc lsnet

Show the network configuration of an HMC:

```
hmc lsnet [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

15. hmc lsresourcerole

Display a resource role of an HMC in detail:

```
hmc lsresourcerole [-v] <hmc> <resourcerole>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

16. hmc lsresourceroles

Show all resource roles of an HMC:

```
hmc lsresourceroles [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

17. hmc lsroute

Show the routing configuration of an HMC:

```
hmc lsroute [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

18. hmc lssysconn

List the state of all connections between an HMC and the attached managed systems:

```
hmc lssysconn [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

19. hmc lstaskrole

Display a task role of an HMC in detail:

```
hmc lstaskrole [-v] <hmc> <taskrole>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

20. hmc lstaskroles

Show all task roles of an HMC:

```
hmc lstaskroles [-v] <hmc>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

21. hmc mkhmcusr

Create a new user account on an HMC:

```
hmc mkhmcusr [-v] <hmc> <user> [<attribute>=<value> …]
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

22. hmc mkresourcerole

Create a new resource role on an HMC:

```
hmc mkresourcerole [-v] <hmc> <resourcerole>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

The newly created resource role is empty.

23. hmc mktaskrole

Create a new task role on an HMC:

```
hmc mktaskrole [-v] <hmc> <taskrole> <parent>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

The *parent* task role restricts the resources which can be assigned to the new task role.

24. hmc remove

Derigistering (removing) an HMC:

```
hmc remove <hmc>
```

When the last HMC of a managed system is removed, the managed systems and LPARs of the managed system are also de-registered and can no longer be administered with the LPAR tool.

25. hmc rescan

Re-capture all HMCs, managed systems and LPARs to match the local mapping files:

```
hmc rescan
```

LPARs created by using the GUI, or LPARs moved by LPM, are no longer correctly recorded in the local mapping files. A rescan updates the local mapping files so that they match the current state again.

26. hmc rmhmcusr

Removing a user on an HMC:

```
hmc rmhmcusr [-v] <hmc> <user>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

27. hmc rmlic

Removing one or more system firmware files on the HMC:

```
hmc rmlic [-v] <hmc> {all|<ppppprrr>|<ppppprrr_lll>}
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

28. hmc rmresourcerole

Removing a resource role:

```
hmc rmresourcerole [-v] <hmc> <resourcerole>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

29. hmc rmtaskrole

Removing a task role:

```
hmc rmtaskrole [-v] <hmc> <taskrole>
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

30. hmc show

Showing registered HMCs with some additional informations:

```
hmc show [<hmc> …]
```

31. hmc shutdown

Shutdown of a HMC:

```
hmc shutdown [-r] [-v] <hmc> [<minutes>]
  -r        # Reboot der HMC
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

32. hmc termtask

Terminating a task (web oder SSH) on the HMC:

```
hmc termtask [-r ssh|webui] [-v] <hmc> <task_id|pid>
  -r ssh    # einen SSH-Task beenden (default)
  -r webui  # einen Web-Task beenden
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

33. hmc version

To show the installed HMC software version:

```
hmc version [-v] [<hmc> …]
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

B. Commandreference - *ms*

The *ms* command performs operations on managed systems.

The version of the LPAR tools can be shown using the option „*-v*“:

```
ms -V
```

1. ms addvswitch

To create a new VSwitch:

```
ms [-h <hmc>] addvswitch [-v] <ms> <switch-name>
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

2. ms bkprofdata

Backing up the profile data of a managed system:

```
ms [-h <hmc>] bkprofdata [-v] <ms> <file>
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

3. ms chled

To turn on or off the status LED of a managed system:

```
ms [-h <hmc>] chled [-v] <ms> off
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

4. ms chlparutil

To change the interval of the processor utilization data assembled:

```
ms [-h <hmc>] chlparutil [-v] <ms> <sample-rate>
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

5. ms chmem

To change the memory configuration of a managed system:

```
ms [-h <hmc>] chmem [-v] <ms> <attributes> …
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

6. ms chprocpool

To change the configuration of a shared processor pool:

```
ms [-h <hmc>] chprocpool [-v] <ms> <pool-name>|<pool-id> <max_pool_proc_unis>
[<reserved_pool_proc_units>]
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

7. ms chsriov

To change the configuration of an SRIOV adapter to *dedicated* or *shared*:

```
ms [-h <hmc>] chsriov [-v] <ms> <slot> {dedicated|shared}
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

8. ms help

To show the available help:

```
ms help [<keyword>]
```

9. ms history

To show status or configuration history of a managed system:

```
ms [-h <hmc>] history [-c] [-v] <ms>
  -h <hmc>      # Hardware Management Console
  -c            # Konfigurations-Änderungen anzeigen
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

10. ms hw

Show hardware components like processors, DIMMs, power supplies an fans:

```
ms [-h <hmc>] hw [-v] <ms>
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

11. ms list

To list the names of managed systems:

```
ms [-h <hmc>] list
  -h <hmc>      # Hardware Management Console
```

12. ms lsled

To show the state of the status LEDs of managed systems:

```
ms [-h <hmc>] lsled [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

13. ms lslic

To show the installed firmware versions of managed systems:

```
ms [-h <hmc>] lslic [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

14. ms lslparutil

To show the utilization interval of managed systems:

```
ms [-h <hmc>] lslparutil [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

15. ms lsprocpool

To show shared processor pools:

```
ms [-h <hmc>] lsprocpool [-f] [-F <attribute-names>] [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -f            # Ausgabe im Stanza-Format
  -F <attribute-names> # Liste der auszugebenden Attribute
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

16. ms lsproprofiles

To show system profiles of managed systems:

```
ms [-h <hmc>] lsprofiles [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -f            # Ausgabe im Stanza-Format
```

```
  -F <attribute-names> # Liste der auszugebenden Attribute
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

17. ms lsprofspace

To show the space used by profiles on managed systems:

```
ms [-h <hmc>] lsprofspace [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

18. ms lsvswitch

To show the virtual ethernet switches of managed systems:

```
ms [-h <hmc>] lsvswitch [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

19. ms memory

To show the memory configuration of managed systems:

```
ms [-h <hmc>] memory|mem [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

20. ms procs

To show processor cores and processor units of managed systems:

```
ms [-h <hmc>] procs [-f] [-F <attribute-names>] [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -f            # Ausgabe im Stanza-Format
  -F <attribute-names> # Liste der auszugebenden Attribute
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

21. ms procstat

To show the processor utilization of a managed system:

```
ms [-h <hmc>] procstat [-n <number>] [-v] <ms>
  -h <hmc>      # Hardware Management Console
  -n <number>   # Anzahl aufzulistende Einträge
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

22. ms properties

To display general properties of managed systems:

```
ms [-h <hmc>] properties|prop [-f] [-F <attribute-names>] [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -f            # Ausgabe im Stanza-Format
  -F <attribute-names> # Liste der auszugebenden Attribute
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

23. ms rmprofile

To remove a system profile for a managed system:

```
ms [-h <hmc>] -s <system-profile> rmprofile|rmprof [-v] <ms>
  -s <system-profile> # das zu löschende System-Profil
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

24. ms rmprofdata

To remove a backup of system profile data of a managed system:

```
ms [-h <hmc>] rmprofdat [-v] <ms> <file>
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

25. ms rmvswitch

To remove a virtual ethernet switch:

```
ms [-h <hmc>] rmvswitch [-v] <ms> <switch-name>
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

26. ms rstprofdata

To restore a managed system from the backed up profile data:

```
ms [-h <hmc>] rstprofdat [-v] <ms> <restore-type> [<file>]
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

27. ms show

To show managed systems with some details:

```
ms [-h <hmc>] show [<ms> …]
  -h <hmc>      # Hardware Management Console
```

28. ms slots

To display the physical slots of a managed system:

```
ms [-h <hmc>] slots [-f] [-F <attribute-names>] [-v] <ms>
  -h <hmc>      # Hardware Management Console
  -f            # Ausgabe im Stanza-Format
  -F <attribute-names> # Liste der auszugebenden Attribute
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

29. ms status

To show the status of managed systems:

```
ms [-h <hmc>] stat|status [-v] [<ms> …]
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

30. ms units

To show I/O units connected to a managed system:

```
ms [-h <hmc>] units [-f] [-F <attribute-names>] [-v] <ms>
  -h <hmc>      # Hardware Management Console
  -f            # Ausgabe im Stanza-Format
  -F <attribute-names> # Liste der auszugebenden Attribute
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

C. Commandreference - *lpar*

All operations related to LPARs are performed with the *lpar* command.

The version of the LPAR tool can be displayed again with the option "*-V*":

```
lpar -V
```

1. lpar activate

An LPAR can be activated using the keyword „*activate*":

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] activate [-b norm|dd|ds|of|sms] [-c] [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # das zu verwendende LPAR-Profil
  -b norm       # normaler Bootmode
  -b dd         # debugging Bootmode
  -b ds         # debugging Bootmode SMS
  -b of         # in den OpenFirmware Mode booten
  -b sms        # im SMS Menü starten
  -c            # eine Konsole für die LPAR starten
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

2. lpar addeth

Adding a virtual ethernet adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addeth [-d] [-i] [-r] [-s <vswitch>] [-t <prio>]
[-v] <lpar> <slot> <pvid> [<vlan>,…]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -i            # IEEE-Mode, erlaubt neben der PVID die Angabe weiterer VLANs
  -r            # der Adapter wird als required konfiguriert
  -s <vswitch>  # der Adapter verwendet den angegebenen VSwitch
  -t <prio>     # Adapter wird als Trunking Adapter mit angegebener Priorität konfiguriert
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
<pvid> : die PVID (Port VLAN ID), nicht-getaggte Pakete wird die PVID zugeordnet
<vlan> : die ID weiterer VLANs, falls die Option -i verwendet wurde
```

3. lpar addfc

Adding a virtual FC client adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addfc [-c] [-d] [-r] [-v] <lpar> [<slot>] <remote-
lpar> [<remote-slot>] [<wwpn1>,<wwpn2>]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
```

```
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -c            # Änderung nur auf dem Client durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -r            # der Adapter wird als required konfiguriert
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
<remote-lpar> : Name der Remote-LPAR
<remote-slot> : die Nummer des virtuellen Slots auf der Remote-LPAR
<wwpn1> : die zu verwendende erste WWPN für den virtuellen Adapter
<wwpn2> : die zu verwendende zweite WWPN für den virtuellen Adapter
```

By default, a server adapter is always added to the virtual I/O server in addition to the client adapter. Using the *-c* option prevents the creation of a virtual FC server adapter on the virtual I/O server. Slot numbers can be optionally specified. If these are not specified, then the LPAR tool automatically selects free slot numbers.

By default, WWPNs are automatically assigned by the managed system. WWPNs can only be assigned to client adapters.

4. lpar addmem

Expand the memory of an LPAR:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addmem [-d] [-v] <lpar> <memory>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<memory> : Die Größe des Hauptspeichers der hinzugefügt werden soll (in MB)
```

5. lpar addprocs

Add processor cores to an LPAR:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addprocs [-d] [-v] <lpar> <procs>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<procs> : Die Anzahl der Prozessoren (Cores) um die erweitert werden soll
```

The number of processors (cores) can only be changed within the limits of minimum and maximum number of processors determined by the current profile of an LPAR.

6. lpar addprocunits

Adding processor units:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addprocunits [-d] [-v] <lpar> <procunits>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<procunits> : Die Anzahl der Prozessor Units um die erweitert werden soll
```

The number of processor units can only be changed within the limits of minimum and maximum number of processor units determined the current profile of the LPAR.

7. lpar addscsi

Adding a virtual SCSI client adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addscsi [-c] [-d] [-r] [-v] <lpar> [<slot>]
<remote-lpar> [<remote-slot>]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -c            # Änderung nur auf dem Client durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -r            # der Adapter wird als required konfiguriert
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
<remote-lpar> : Name der Remote-LPAR
<remote-slot> : die Nummer des virtuellen Slots auf der Remote-LPAR
```

By default, a server adapter is always added to the virtual I/O server in addition to the client adapter. Using the *-c* option prevents the creation of a VSCSI server adapter on the virtual I/O server. Slot numbers can be optionally specified. If these are not specified, then the LPAR tool automatically selects free slot numbers.

8. lpar addserial

Adding a virtual serial adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addserial [-c] [-d] [-r] [-v] <lpar> <slot>
<remote-lpar> <remote-slot>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -c            # einen Client Adapter hinzufügen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -r            # der Adapter wird als required konfiguriert
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
<remote-lpar> : Name der Remote-LPAR
<remote-slot> : die Nummer des virtuellen Slots auf der Remote-LPAR
```

9. lpar addslot

Adding a physical slot:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] addslot [-d] [-f] [-r] [-v] <lpar> <drc-index>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -f            #
  -r            # der Adapter wird als required konfiguriert
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<drc-index> : der drc-index des Slots, wie er bei „ms slots“ angezeigt wird
```

10. lpar chmem

Changing memory attributes:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] chmem [-d] [-v] [-x <expansion>] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
  -x <expansion> # AME-Faktor ändern
```

11. lpar console

Opening a console for an LPAR:

```
lpar [-h <hmc>] [-m <ms>] console [-f] [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -f            # „force“ das Öffnen der Konsole erzwingen
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

12. lpar create

Creating a new LPAR:

```
lpar [-h <hmc>] -m <ms> [-p <profile>] create [-i <lparid>] [-t <template>] [-v] [<lpar>]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Name des LPAR-Profils für die neue LPAR, default ist standard
  -i <lparid>   # die LPAR-ID für die neue LPAR
  -t <template> # Template-File mit dem die LPAR erzeugt werden soll
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

The LPAR name is optional, if it is not specified, an LPAR name is generated (*lparN*). The managed system on which the LPAR is to be created must be specified!

13. lpar delete

Deleting an LPAR:

```
lpar [-h <hmc>] [-m <ms>] delete [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

14. lpar help

Display available help:

```
lpar help [<keyword>|eth|fc|scsi|usage]
```

15. lpar list

List the names of LPARs:

```
lpar [-h <hmc>] [-m <ms>] list [<lpar> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
```

If no LPARs are specified as arguments, then all known LPARs will be listed. If a managed system is specified, only LPARs of this managed system will be listed.

16. lpar lsrefcode

List reference codes for an LPAR:

```
lpar [-h <hmc>] [-m <ms>] lsrefcode [-n <number>] [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -n <number>   # Anzahl aufzulistende Codes, default ist 10
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

17. lpar memory

Display the memory configuration of LPARs:

```
lpar [-h <hmc>] [-m <ms>] mem|memory [-v] [<lpar> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

18. lpar migrate

Perform a LPM (Live Partition Migration) operation:

```
lpar [-h <hmc>] [-m <ms>] migrate [-v] <lpar> <target-ms>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<target-ms> : Das Ziel Managed System für die Migration
```

19. lpar mkblueprint

Generate a blueprint from an existing LPAR:

```
lpar [-h <hmc>] [-m <ms>] mkblueprint [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

20. lpar oslevel

Display the version of the operating system of LPARs:

```
lpar [-h <hmc>] [-m <ms>] oslevel [-v] [<lpar> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

21. lpar procs

Display the processor configuration of LPARs:

```
lpar [-h <hmc>] [-m <ms>] procs [-v] [<lpar> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

22. lpar rename

Rename an LPAR:

```
lpar [-h <hmc>] [-m <ms>] rename [-v] <lpar> <new-lpar-name>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

23. lpar rmeth

Remove a virtual ethernet adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmeth [-d] [-v] <lpar> <slot>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
```

24. lpar rmfc

Remove a virtual FC adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmfc [-c] [-d] [-v] <lpar> <slot>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -c            # Änderung nur auf dem Client durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
```

25. lpar rmmem

Reduce the amount of memory of an LPAR:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmmem [-d] [-v] <lpar> <memory>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<memory> : Die Größe des Hauptspeichers der weggenommen werden soll (in MB)
```

26. lpar rmprocs

Remove processor (cores) from an LPAR:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmprocs [-d] [-v] <lpar> <procs>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<procs> : Die Anzahl der Prozessoren (Cores) die weggenommen werden sollen
```

The number of processors (cores) can only be changed within the limits of minimum and maximum number of processors determined by the current profile.

27. lpar rmprocunits

Remove processor units from an LPAR:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmprocunits [-d] [-v] <lpar> <procunits>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<procunits> : Die Anzahl der Prozessor Units die weggenommen werden sollen
```

The number of processor units can only be changed within the limits of minimum and maximum number of processor units determined by the current profile.

28. lpar rmscsi

Remove a virtual SCSI adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmscsi [-c] [-d] [-v] <lpar> <slot>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -c            # Änderung nur auf dem Client durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
```

29. lpar rmserial

Remove a virtual serial adapter:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmserial [-d] [-v] <lpar> <slot>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<slot> : die Nummer des virtuellen Slots für den Adapter
```

30. lpar rmslot

Remove a physical slot:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] rmslot [-d] [-v] <lpar> <drc-index>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System der LPAR
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<drc-index> : der drc-index des Slots, wie er bei „ms slots“ angezeigt wird
```

31. lpar show

Display LPARs with some additional informations:

```
lpar [-h <hmc>] [-m <ms>] show [<lpar> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
```

If no LPARs are specified as arguments, then all known LPARs will be listed. If a managed system is specified, only LPARs of this managed system will be listed.

32. lpar shutdown

Shutting down an LPAR:

```
lpar [-h <hmc>] [-m <ms>] shutdown [-f] [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -f            # Herunterfahren erzwingen, die LPAR wird einfach ausgeschaltet
```

33. lpar slots

Displaying the physical slot of an LPAR:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] slots [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -p <profile>  # das LPAR-Profil das angezeigt werden soll
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

34. lpar status

Displaying the status of LPARs:

```
lpar [-h <hmc>] [-m <ms>] stat|status [-v] [<lpar> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

35. lpar validate

Validate an LPM (Live Partition Migration) operation:

```
lpar [-h <hmc>] [-m <ms>] validate [-v] <lpar> <target-ms>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen

<target-ms> : Das Ziel Managed System für die Validierung
```

36. lpar vslots

Display the virtual slots of an LPAR:

```
lpar [-h <hmc>] [-m <ms>] [-p <profile>] vslots [-v] <lpar>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -p <profile>  # das LPAR-Profil das angezeigt werden soll
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

D. Commandreference - *vios*

All operations involving the administration of a virtual I/O server, are performed with the command *vios*. Logging in directly on the virtual I/O server is usually no longer necessary.

The version of the LPAR tool can be shown using the option „-V“:

```
vios -V
```

1. vios addfc

Adding a virtual FC server adapter on a virtual I/O server:

```
vios [-h <hmc>] [-m <ms>] [-p <profile>] addfc [-d] [-r] [-v] <vios> <slot> <remote-lpar>
<remote-slot>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -r            # Adapter ist benötigt zur LPAR Aktivierung
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

2. vios addscsi

Adding a virtual SCSI server adapter on a virtual I/O server:

```
vios [-h <hmc>] [-m <ms>] [-p <profile>] addscsi [-d] [-r] [-v] <vios> <slot> <remote-lpar>
<remote-slot>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -p <profile>  # Änderung nur in diesem LPAR-Profil durchführen
  -d            # Änderung nur dynamisch durchführen (DLPAR-Operation)
  -r            # Adapter ist benötigt zur LPAR Aktivierung
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

3. vios chrep

Changing the size of the virtual media repository:

```
vios [-h <hmc>] [-m <ms>] chrep [-v] <vios> <additional-size>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

4. vios help

Display the available help:

```
vios help [<keyword>|device|dlpar|media|sp|usage]
```

5. vios ioslevel

Show the IOS version of the virtual I/O servers:

```
vios [-h <hmc>] [-m <ms>] ioslevel [-v] [<vios> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

6. vios list

List the virtual I/O servers:

```
vios [-h <hmc>] [-m <ms>] list [<vios> …]
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
```

Only the names of the virtual I/O servers are listed.

7. vios loadopt

To load a medium into a virtual optical drive:

```
vios [-h <hmc>] [-m <ms>] loadopt [-f] [-r] [-v] <vios> <vtopt> <media>
  -h <hmc>      # Hardware Management Console
  -m <ms>       # das Managed System
  -f
  -r
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
<vtopt>: das virtuelle optische Laufwerk
<media>: das einzulegende Medium
```

8. vios lsnports

Display the NPIV ports of virtual I/O servers:

```
vios [-h <hmc>] [-m <ms>] lsnports [-v] [<vios> …]
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

9. vios lsrep

List the virtual media repository:

```
vios [-h <hmc>] [-m <ms>] lsrep [-v] <vios>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

10. vios lssp

List this storage pools on a virtual I/O servers:

```
vios [-h <hmc>] [-m <ms>] lssp [-v] <vios> [<sp>]
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
<sp>            # der Storage Pool
```

If no storage pool is specified, all existing storage pools are listed. If a storage pool is specified, the storage pools with the included backing devices are listed.

11. vios lsvopt

Display the virtual optical media:

```
vios [-h <hmc>] [-m <ms>] lsvopt [-v] <vios>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

12. vios map

Map a *hdisk* or logical volume to a *vhost* adapter:

```
vios [-h <hmc>] [-m <ms>] map [-v] <vios> <vhost> <target-device> [<vtd>]
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

13. vios mkbdsp

Create a backing device in a storage pool and optionally assign it to a *vhost* adapter:

```
vios [-h <hmc>] [-m <ms>] mkbdsp [-v] <vios> <sp> <backing-device> <size> [<vhost> [<vtd>]]
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
  <sp>          # der Storage Pool
  <backing-device> # Name des zu erzeugenden Backing Devices
  <size>        # Größe des Backing Devices
  <vhost>       # vhost-Adapter
  <vtd>         # Name des Virtual Target Devices
```

14. vios mkfbo

Create a virtual optical drive:

```
vios [-h <hmc>] [-m <ms>] mkfbo [-v] <vios> <vhost> [<vtd>]
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
  <vhost>       # vhost-Adapter
  <vtd>         # Name des Virtual Target Devices
```

15. vios mkrep

Create a virtual media repository:

```
vios [-h <hmc>] [-m <ms>] mkrep [-v] <vios> <size>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

16. vios npiv

Display the NPIV mappings:

```
vios [-h <hmc>] [-m <ms>] npiv [-v] <vios>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

17. vios rmbdsp

Remove a backing device:

```
vios [-h <hmc>] [-m <ms>] rmbdsp [-v] <vios> <sp> <backing-device>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
  <sp>          # zu löschende Backing Device
  <backing-device>  # Name des zu löschenden Backing Devices
```

18. vios rmfc

Remove a virtual FC server adapter:

```
vios [-h <hmc>] [-m <ms>] [-p <profile>] rmfc [-d] [-v] <vios> <slot>
  -m <ms>       # das Managed System
```

```
  -h <hmc>      # Hardware Management Console
  -p <profile>  # Adapter nur in diesem Profil löschen
  -d            # Löschen nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

By default the adapter will be removed dynamically and in the current profile.

19. vios rmrep

Remove the virtual media repository:

```
vios [-h <hmc>] [-m <ms>] rmrep [-v] <vios>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

20. vios rmscsi

Remove a virtual SCSI server adapter:

```
vios [-h <hmc>] [-m <ms>] [-p <profile>] rmscsi [-d] [-v] <vios> <slot>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -p <profile>  # Adapter nur in diesem Profil löschen
  -d            # Löschen nur dynamisch durchführen (DLPAR-Operation)
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

By default the adapter will be removed dynamically and in the current profile.

21. vios rmvopt

Removing a media file from the virtual media repository:

```
vios [-h <hmc>] [-m <ms>] rmvopt [-f] [-v] <vios> <media-file>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v            # verbose only, Aktionen anzeigen aber nicht durchführen
```

22. vios sea

Display the SEA (shared ethernet adapters) of a virtual I/O servers:

```
vios [-h <hmc>] [-m <ms>] sea [-v] <vios>
  -m <ms>       # das Managed System
  -h <hmc>      # Hardware Management Console
  -v        # verbose only, Aktionen anzeigen aber nicht durchführen
```

23. vios show

Display virtual I/O server in detail:

```
vios [-h <hmc>] [-m <ms>] show [<vios> …]
  -h <hmc>       # Hardware Management Console
  -m <ms>        # das Managed System
```

24. vios unloadopt

Unload the medium from a virtual optical drive:

```
vios [-h <hmc>] [-m <ms>] unloadopt [-r] [-v] <vios> <vtopt>
  -h <hmc>       # Hardware Management Console
  -m <ms>        # das Managed System
  -r
  -v             # verbose only, Aktionen anzeigen aber nicht durchführen
<vtopt>: das virtuelle optische Laufwerk
```

25. vios unmap

Unmap a virtual target device from a *vhost* adapter:

```
vios [-h <hmc>] [-m <ms>] unmap [-v] <vios> <vtd>
  -m <ms>        # das Managed System
  -h <hmc>       # Hardware Management Console
  -v         # verbose only, Aktionen anzeigen aber nicht durchführen
```

26. vios vfcmap

Map a *vfchost* adapter to a physical FC port:

```
vios [-h <hmc>] [-m <ms>] vfcmap [-v] <vios> <vfchost> [<fcs>]
  -h <hmc>       # Hardware Management Console
  -m <ms>        # das Managed System
  -v         # verbose only, Aktionen anzeigen aber nicht durchführen
```

27. vios vscsi

Display informations for the VSCSI server adapters:

```
vios [-h <hmc>] [-m <ms>] vscsi [-v] <vios> [<vhost>]
  -h <hmc>       # Hardware Management Console
  -m <ms>        # das Managed System
  -v             # verbose only, Aktionen anzeigen aber nicht durchführen
```

E. Configurationparameters

The */opt/pwrcmps/etc/lpar.cfg* and *~/.lpar.cfg* files can be used to configure the LPAR tool. The system-wide configuration file (if available) is read in first. The user-specific configuration file (if available) is read in afterwards. If a parameter is set in both files, the last user-specific configuration applies.

ConfigDirectory:

This parameter determines where the *hmc.list*, *ms.list*, and *lpar.list* files are created.

Default: the user's home directory.

LicenseFile:

The parameter determines where the license file is located. For installations with *tar*-file in the home directory of a user, this parameter must be set.

Default: */opt/pwrcmps/etc/lpar.lic* and *~/.lpar.lic*.

ControlPersist:

Specifies how long an inactive SSH master connection is kept open.

Default: 5 minutes.

ServerAliveInterval:

Time interval within which an alive message is expected from the server.

Default: 10 seconds.

ServerAliveCountMax:

Maximum number of outstanding alive messages from the server.

Default: 2.

LowestVirtualClientSlot:

Lowest allowed slot number for a virtual client adapter.

Default: 10.

LowestVirtualServerSlot:

Lowest allowed slot number for a virtual server adapter.

Default: 20.

HmcUsageShow:

List of keywords displayed on usage output of the *hmc* command.

Default: all.

LparUsageShow:

List of keywords displayed on usage output of the *lpar* command.

Default: all.

MsUsageShow:

List of keywords displayed on usage output of *ms* command.

Default: all.

ViosUsageShow:

List of keywords displayed on usage issues of the *vios* command.

Default: all.