The s390-tools package in a nutshell

Susanne Wintenberger (swinten@de.ibm.com)
IBM Lab Boeblingen, Germany
Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

Not all common law marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within its relevant market. Those trademarks followed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml:

* AS/400®, e-business(logo)®, DBE, ESCO, eServer, FICON, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System x, System z, System z9®, BladeCenter®

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply. All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions. This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the Performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

IBM and Linux: Community Innovation for your Business
Agenda

* Introduction
  - What is the s390-tools package
  - Contained applications
* Selected Tools
  - Shutdown Action Tools
  - Chzcrypt / Lszcrypt
  - IUCV terminal applications
  - Cpuplugd
  - Chchp / Lschp
  - DASD, Debug & Dump
  - Lslun / ....
  - z/VM related tools
* Where to find more Information
What is the s390-tools package?

- s390-tools is a package with a set of user space utilities to be used with the Linux on System z distributions.
- It is the essential tool chain for Linux on System z
- It contains everything from the boot loader to dump related tools for a system crash analysis.
- Version 1.8.1 and was released in May 2009 and latest version is 1.8.2, released in September 2009
- This software package is contained in all major (and IBM supported) distributions which support s390
  - RedHat Enterprise Linux 4
  - RedHat Enterprise Linux 5
  - SuSE Linux Enterprise Server 10
  - SuSE Linux Enterprise Server 11
- Feedback: linux390@de.ibm.com
The Content

- **CHANGE**
  - chccwdev
  - chchk
  - chreipl
  - chshut
  - chzcrypt

- **DASD**
  - dasdfmt
  - dasdinfo
  - dasdview
  - fdasd
  - tunedasd
  - mon_fsstatd
  - mon_procd
  - monmon

- **DISPLAY**
  - lcss
  - lschp
  - lsdasd
  - lsluns
  - lsqeth
  - lsreipl
  - lsshut
  - lstape
  - lszcrypt
  - lszfcp

- **MONITOR**
  - ip_watcher
  - osasnmrd
  - qetharp
  - qethconf
  - znetconf

- **NETWORK**
  - tape390_crypt
  - tape390_display

- **TAPE**
  - dbginfo
  - dumpconf
  - zfcpdump
  - zfcpdbf
  - zgetdump
  - scsi_logging
  - _level

- **DUMP & DEBUG**
  - vmconvert
  - vmcp
  - vmur
  - cpuplugd
  - iucvconn
  - iucvtty
  - ts-shell

- **MISC**
  - zipl

- **BOOT**
  - dasdfmt
  - dasdinfo
  - dasdview
  - fdasd
  - tunedasd
Shutdown action tools

chreipl: Configure a disk as well as select an optional entry in the boot menu for the next boot cycle.

```
root@larsson:~> chreipl node /dev/dasda
root@larsson:~> chreipl node /dev/sda
root@larsson:~> chreipl ccw -d 0.0.7e78 -L 1
root@larsson:~> chreipl fcp --wwpn 0x500507630300c562 --lun 0x401040B300000000 -d 0.0.1700
```
Shutdown action tools (cont.)

chshut: controls the system behavior on shutdown actions

```bash
root@larsson:~> chshut halt ipl
root@larsson:~> chshut halt vmcmd LOGOFF
root@larsson:~> chshut reboot reipl
root@larsson:~> chshut poff vmcmd "MSG MASTER Going down" \ vmcmd "LOGOFF"
```
Shutdown action tools (cont.)

lsreipl: command to see from which device your system will boot after you issue the reboot command. Further you can query the system for information about the current boot device.

root@larsson:~> lsreipl
Re-IPL type: ccw
Device: 0.0.4bb8
Loadparm:
root@larsson:~> lsreipl -i

lsshut: command to see what the system should do in one of the following states.

root@larsson:~> lsshut
Trigger          Action
========================
Halt              stop
Panic             stop
Power off         vmcmd (LOGOFF)
Reboot            reipl
lszcrypt

* Use the lszcrypt command to display information about cryptographic adapters managed by zcrypt and zcrypt’s AP bus attributes

* To display card type and online status of all available cryptographic adapters:
  ```
  root@larsson:~> lszcrypt -V
  ```

* To display card type, online status, hardware card type, hardware queue depth, and request count for cryptographic adapters 0, 1, 10, and 12
  ```
  root@larsson:~> lszcrypt -VV 0 1 10 12
  ```

* To display AP bus information:
  ```
  root@larsson:~> lszcrypt -b
  ```
chzcrypt

* Use the chzcrypt command to configure cryptographic adapters managed by zcrypt and modify zcrypt’s AP bus attributes.

```
root@larsson:~> chzcrypt -e 0 1 4 5 12
```

* To set the cryptographic adapters 0, 1, 4, 5, and 12 online:

```
root@larsson:~> chzcrypt -e 0 1 4 5 12
```

* To set all available cryptographic adapters offline:

```
root@larsson:~> chzcrypt -d -a
```

* To set the configuration timer for re-scanning the AP bus to 60 seconds and disable zcrypt's poll thread:

```
root@larsson:~> chzcrypt -c 60 -n
```
znetconf

* Use the znetconf command to list and configure network devices.

* To list all configured network devices:

```
root@larsson:~> znetconf -c
```

<table>
<thead>
<tr>
<th>Device IDs</th>
<th>Type</th>
<th>Card Type</th>
<th>CHPID</th>
<th>Drv. Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.f500,0.0.f501,0.0.f502</td>
<td>1731/01</td>
<td>OSD_1000</td>
<td>76</td>
<td>qeth</td>
<td>eth0</td>
</tr>
</tbody>
</table>

* To configure the potential network device 0.0.f503 with the layer2 option with the value 0 and the portname option with the value myname:

```
root@larsson:~> znetconf -a f503 -o layer2=0 -o portname=mynname
```
IUVC terminal applications

* Full-screen terminal access to Linux guest operating systems on the same z/VM
* Access Linux instances with no external network because IUCV is independent from TCP/IP
IUCV terminal applications (cont.)

* The IUCV terminal applications consist of:
  - `iucvconn` – Start terminal connection over IUCV
  - `iucvtty` – Allow remote logins over IUCV
  - `ts-shell` – Login shell for terminal servers over IUCV

* Terminal access over IUCV is provided by:
  - `iucvtty`, or
  - z/VM IUCV hypervisor console device driver (Linux kernel)

* For more details, see
  - How to setup a Terminal Server Environment on z/VM (SC34-2596-00)
cpuplugd

* Use the cpuplugd command to:
  - Enable or disable CPUs based on a set of rules. This increases the performance of single threaded applications within a z/VM or LPAR environment with multiple CPUs. The rules can incorporate certain system load variables.
  - Manage memory under z/VM.

* Configuration file: /etc/sysconfig/cpuplugd
* Init-Script: /etc/init.d/cpuplugd {start, stop, restart}
cpuplugd: Example Configuration

**UPDATE**="60"

**CPU_MIN**="2"
**CPU_MAX**="10"

**HOTPLUG** = "(loadavg > onumcpus +0.75) & (idle < 10.0)"
**HOTUNPLUG** = "(loadavg < onumcpus -0.25) | (idle > 50)"

**CMM_MIN**="0"
**CMM_MAX**="8192"
**CMM_INC**="256"

**MEMPLUG** = "swaprate > freemem+10 & freemem+10 < apcr"
**MEMUNPLUG** = "swaprate > freemem + 10000"
Change: chchp

* Use chchp (Change channel path status) to set channel paths online or offline.

```
  chchp
    -c 0
    -v 0
    -a <key>=<value>
```

* The -c option is equivalent to performing a Configure Channel Path Off or Configure Channel Path On operation on the hardware management console.

* To set channel path 0.40 to standby configuration state:

  ```
  root@larsson:~> chchp --configure 0 0.40
  Configure standby 0.40... done.
  ```

* To set the channel path with the channel path ID 0.40 to the configured state issue:

  ```
  root@larsson:~> chchp --configure 1 0.40
  Configure online 0.40... done.
  ```
Change: chchp (cont.)

* To set channel paths 0.65 to 0.6f to the configured state issue:

```
root@larsson:~> chchp -c 1 0.65-0.6f
```

* Use the -v option to change the logical channel path state to online or offline

* To set channel paths 0.12, 0.7f and 0.17 to 0.20 to the logical offline state issue:

```
root@larsson:~> chchp -v 0 0.12,0.7f,0.17-0.20
```

* Use the -a option to change the channel path sysfs attribute (e.g. Configure, status) to a value.

* To set channel path 0.19 into standby state issue:

```
root@larsson:~> chchp -a configure=0 0.19
```
DASD

Large Volume Support is a feature that allows to use ECKD devices with more than 65520 cylinders.
- This allows to have DASDs with more than 45Gigabyte
- This features is available with DS8000 R4.0
- Included in Linux Kernel 2.6.30
- All DASD related tools have been updated for Large Volume Support
  - dasdfmt
  - fdasd
  - dasdview
  - zipl
  - dump tools
Dump: Multi Volume Dump

* How to prepare a set of ECKD DASD devices for a multi-volume dump? (64-bit systems only)
  - You can specify up to 32 ECKD DASD partitions for a multi-volume dump. We use two DASDs in this example:

    root@larsson:~> dasdfmt -f /dev/dasdc -b 4096
    root@larsson:~> dasdfmt -f /dev/dasdd -b 4096

  - Create the partitions with fdasd. The sum of the partition sizes must be sufficiently large (the memory size + 10 MB):

    root@larsson:~> fdasd /dev/dasdc
    root@larsson:~> fdasd /dev/dasdd

  - Create a file called sample_dump_conf containing the device nodes of the two partitions, separated by one or more line feed characters
  - Prepare the volumes using the zipl command.

    root@larsson:~> zipl -M sample_dump_conf
    [...]

...
How to obtain a multi-volume dump

* To obtain a dump with the multi-volume DASD dump tool, perform the following steps:
  - Stop all CPUs, Store status on the IPL CPU.
  - IPL the dump tool using one of the prepared volumes, either 4711 or 4712.
  - After the dump tool is IPLed, you'll see a messages that indicates the progress of the dump. Then you can IPL Linux again

```
#cp cpu all stop
#cp cpu 0 store status
#cp ipl 4711
```

* Copying a multi-volume dump to a file
  - Use zgetdump without any option to copy the dump parts to a file:

```
root@larsson:~> zgetdump /dev/dasdc > mv_dump_file
```
How to obtain information about a multi-volume dumps

* Display information of the involved volumes:

```
root@larsson:~> zgetdump -d /dev/dasdc
'/dev/dasdc' is part of Version 1 multi-volume dump, which is
spread along the following DASD volumes:
0.0.4711 (online, valid)
0.0.4712 (online, valid)
[...]
```

* Display information about the dump itself:

```
root@larsson:~> zgetdump -i /dev/dasdc
Dump device: /dev/dasdc
>>> Dump header information <<<
Dump created on: Fri Aug  7 15:12:41 2009
[...]
Multi-volume dump: Disk 1 (of 2)
Reading dump contents from
0.0.4711..................................
Dump ended on:   Fri Aug  7 15:12:52 2009
Dump End Marker found: this dump is valid.
```
dumpconf

* The dumpconf tool configures a dump device that is used for automatic dump in case of a kernel panic.
  - The command can be installed as service script under `/etc/init.d/dumpconf` or can be called manually.
  - Start service: `# service dumpconf start`
  - It reads the configuration file `/etc/sysconfig/dumpconf`.
  - Example configuration for CCW dump device (DASD) and reipl after dump:

```
ON_PANIC=dump_reipl
DUMP_TYPE=ccw
DEVICE=0.0.4711
```

- Example configuration for FCP dump device (SCSI disk):

```
ON_PANIC=dump
DUMP_TYPE=fcp
DEVICE=0.0.4714
WWPN=0x5005076303004712
LUN=0x4047401300000000
BOOTPROG=0
BR_LBA=0
```
dumpconf (cont.)

* Example configuration for re-IPL without taking a dump, if a kernel panic occurs:

```bash
ON_PANIC=reipl
```

* Example of executing a CP command, and rebooting from device 4711 if a kernel panic occurs:

```bash
ON_PANIC=vmcmd
VMCMD_1="MSG <vmguest> Starting VMDUMP"
VMCMD_2="VMDUMP"
VMCMD_3="IPL 4711"
```
## Dump Tools Summary

<table>
<thead>
<tr>
<th>Tool</th>
<th>Stand alone tools</th>
<th>VMDUMP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>DASD</strong></td>
<td><strong>Tape</strong></td>
</tr>
<tr>
<td>Environment</td>
<td>VM&amp;LPAR</td>
<td>LPAR</td>
</tr>
<tr>
<td>Preparation</td>
<td>Zipl -d /dev/&lt;dump_dev&gt;</td>
<td>Mkdir /dumps/mydumps zipl -D /dev/sda1 ...</td>
</tr>
<tr>
<td>Creation</td>
<td>Stop CPU &amp; Store status ipl &lt;dump_dev_CUU&gt;</td>
<td>Vmdump</td>
</tr>
<tr>
<td>Dump medium</td>
<td>ECKD or FBA</td>
<td>Tape cartridges</td>
</tr>
</tbody>
</table>
| Copy to filesystem | Zgetdump /dev/<dump_dev> > dump_file | --- | Dumpload ftp ... vmconvert ...
| Viewing            | Lcrash or crash                    |        |         |

dbginfo.sh

* dbginfo.sh is a script to collect various system related files, for debugging purposes. It generates a tar-archive which can be attached to PMRs / Bugzilla entries
* It is similar to the RedHat tools sosreport

root@larsson:~> **dbginfo.sh**
Create target directory /tmp/DBGINFO-2009-04-15-22-06-20-t6345057
Change to target directory /tmp/DBGINFO-2009-04-15-22-06-20-t6345057
Get procfs entries
Saving runtime information into runtime.out
Get file list of /sys
Get entries of /sys
[...]
**DISPLAY: Isluns**

* Use the Isluns command to discover and scan LUNs in Fibre Channel Storage Area Networks (SANs).
  - This example shows all LUNs for port 0x500507630300c562:

    ```
    root@larsson:~> lsluns --port 0x500507630300c562
    Scanning for LUNs on adapter 0.0.5922
    at port 0x500507630300c562:
    0x4010400000000000
    0x4010400100000000
    [...]
    ```

  - This example shows all LUNs for adapter 0.0.5922:

    ```
    root@larsson:~> lsluns -c 0.0.5922
    at port 0x500507630300c562:
    0x4010400000000000
    0x4010400100000000
    [...]
    at port 0x500507630303c562:
    0x4010400000000000
    [...]
    ```
z/VM: vmcp

* Using the z/VM CP interface device driver (vmcp), you can send control program (CP) commands to the VM hypervisor and display VM’s response.

```
root@larsson:~> modprobe vmcp
root@larsson:~> vmcp "q dasd"|grep T6345057
DASD 4DE0 ATTACHED T0 T6345057 4DE0 R/W 0X4DE0
DASD 4DE1 ATTACHED T0 T6345057 4DE1 R/W 0X4DE1
DASD 4DE2 ATTACHED T0 T6345057 4DE2 R/W 0X4DE
DASD 4DE3 ATTACHED T0 T6345057 4DE3 R/W 0X4DE3
```
The `vmur` command provides all functions required to work with z/VM spool file queues:
- **Receive**: Read data from the z/VM reader file queue
- **Punch or print**: Write data to the z/VM punch or printer file queue and transfer it to another user’s virtual reader, optionally on a remote z/VM node.
- **List**: Display detailed information about one or all files on the specified spool file queue.
- **Purge**: Remove one or all files on the specified spool file queue.
- **Order**: Position a file at the top of the specified spool file queue.
vmur (cont.)

* Produce and read Linux guest machine dump
  - Produce guest machine dump:

```shell
root@larsson:~> vmcp vmdump
```

  - Find spool ID of VMDUMP spool file in the output of the vmur li command:

```shell
root@larsson:~> vmur li
```

```
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE DIST
T6360025 0463 V DMP 00020222 001 NONE 06/11 15:07:42
VMDUMP FILE T6360025
```

  - Move vmdump file to top of reader queue with the vmur order command: # vmur or 463

```shell
root@larsson:~> vmur or 463
```

  - Read and convert the vmdump file to a file (lkcd dump format) on the Linux file system in the current working directory and close the virtual reader

```shell
root@larsson:~> chccwdev -e 000c
root@larsson:~> vmconvert /dev/vmrdr-0.0.000c linux_dump
root@larsson:~> vmcp cl c
```
vmur (cont.)

* Log and read Linux guest machine console
  - Begin console spooling:

    root@larsson:~> vmcp sp cons start

    - Produce output to VM console (for example, with CP TRACE).
    - Close the console file and transfer it to the reader queue, find the spool ID behind the FILE keyword in the corresponding CP message.

    root@larsson:~> vmcp sp cons clo \* rdr
    RDR FILE 0398 SENT FROM T6360025 CON WAS 0398 RECS 1872 CPY 001 T NOHOLD NOKEEP

    - Read the guest machine console file into a file on the Linux file system in the current working directory:

    root@larsson:~> chccwdev -e 000c
    root@larsson:~> vmur re -t 398 linux_cons
vmur (cont.)

* Prepare z/VM reader to IPL Linux image
  - Send parmfile to VM punch and transfer it to the reader queue and find the parmfile spool id message

```
root@larsson:~> vmur pun -r /boot/parmfile
[...]
Reader file with spoolid 0465 created.
```

  - Send image to VM punch and transfer it to reader queue:

```
root@larsson:~> vmur pun -r /boot/vmlinuz -N image
```

  - Move image to first and parmfile to the second position in the reader queue:

```
root@larsson:~> vmur or 465
root@larsson:~> vmur or 466
```

  - Prepare re-IPL from the VM reader and boot the Linux image

```
root@larsson:~> chreipl ccw 0.0.000c
root@larsson:~> reboot
```
When a Linux on System z instance boots, it senses and analyses all available devices. You can use the cio_ignore kernel parameter to specify a list of devices that are to be ignored. The following applies to ignored devices:

- Ignored devices are not sensed and analyzed. The device cannot be used unless it has been analyzed.
- Ignored devices are not represented in sysfs.
- Ignored devices do not occupy storage in the kernel.
- The subchannel to which an ignored device is attached is treated as if no device were attached.
- cio_ignore might hide essential devices such as the console under z/VM. The console is typically device number 0.0.0009.

This example specifies that all devices in the range 0.0.b100 through 0.0.b1ff, and the device 0.0.a100 are to be ignored.

cio_ignore=0.0.b100-0.0.b1ff,0.0.a100
cio_ignore (cont.)

* Display ignored devices:

```bash
root@larsson:~> cat /proc/cio_ignore
0.0.0000-0.0.78ff
0.0.f503-0.0.ffff
```

* Free an individual device from the ignore list

```bash
root@larsson:~> echo free 0.0.4711 >/proc/cio_ignore
```

* Free all devices from the ignore list

```bash
root@larsson:~> echo free all >/proc/cio_ignore
```
More Information

Linux on System z

Using the Dump Tools
November, 2008

Device Drivers, Feature
November, 2008

How to Set up a Terminal Server
Environment on z/VM
June 2009

More Information

Questions?