

Linux on z Systems and LinuxONE



# Using the Dump Tools on Red Hat Enterprise Linux 7.4



Linux on z Systems and LinuxONE



# Using the Dump Tools on Red Hat Enterprise Linux 7.4

**Note**

Before using this information and the product it supports, read the information in "Notices" on page 63.

This edition applies to Red Hat Enterprise Linux 7.4 on IBM System z, and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2014, 2017.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Summary of changes</b> . . . . .	<b>v</b>
Updates for Red Hat Enterprise Linux 7.3 and 7.4 . . . . .	v
Updates for Red Hat Enterprise Linux 7.2 . . . . .	v
<b>About this book</b> . . . . .	<b>vii</b>
Other relevant Linux on IBM z Systems publications	vii
<b>Chapter 1. Planning for dumps</b> . . . . .	<b>1</b>
Tools overview . . . . .	1
kdump . . . . .	2
Stand-alone tools . . . . .	2
The VMDUMP tool . . . . .	3
Live-system dump . . . . .	3
Maximum dump size by tool . . . . .	3
Dump methods compared . . . . .	4
Automatic dumping . . . . .	5
Testing automatic dump-on-panic . . . . .	5
Sharing dump devices between different versions of Linux . . . . .	6
<b>Chapter 2. Using kdump</b> . . . . .	<b>7</b>
How kdump works on z Systems . . . . .	7
Setting up kdump . . . . .	9
Initiating a dump . . . . .	10
z/VM guest example . . . . .	10
LPAR example . . . . .	11
<b>Chapter 3. Using a DASD dump device</b> <b>13</b>	<b>13</b>
Installing the DASD dump tool . . . . .	13
Initiating a DASD dump . . . . .	14
z/VM guest example . . . . .	14
Dumping an LPAR using the HMC or SE web interface example . . . . .	15
Automatic dump example . . . . .	16
Copying the dump from DASD with zgetdump . . . . .	17
<b>Chapter 4. Using DASD devices for multi-volume dump</b> . . . . .	<b>19</b>
Installing the multi-volume DASD dump tool . . . . .	20
Initiating a multi-volume DASD dump . . . . .	21
Copying a multi-volume dump to a file . . . . .	22
<b>Chapter 5. Using a tape dump device</b> <b>23</b>	<b>23</b>
Installing the tape dump tool . . . . .	23
Initiating a tape dump . . . . .	23
z/VM guest example . . . . .	24
LPAR example . . . . .	25
Copying the dump from tape . . . . .	27
Preparing the dump tape . . . . .	27
Using the zgetdump tool to copy the dump . . . . .	27
Checking whether a dump is valid, and printing the dump header . . . . .	28
<b>Chapter 6. Using a SCSI disk dump device</b> . . . . .	<b>29</b>
Installing the SCSI disk dump tool . . . . .	29
SCSI dump tool parameters . . . . .	29
Combined dump and target partition . . . . .	30
Initiating a SCSI dump . . . . .	31
z/VM guest example . . . . .	32
LPAR example . . . . .	33
Automatic dump example . . . . .	35
Printing the SCSI dump header . . . . .	36
<b>Chapter 7. Creating dumps on z/VM with VMDUMP</b> . . . . .	<b>37</b>
Initiating a dump with VMDUMP . . . . .	37
z/VM guest example . . . . .	37
Copying the dump to Linux . . . . .	38
<b>Chapter 8. Creating live-system dumps with zgetdump</b> . . . . .	<b>39</b>
Creating a kernel dump on a live system . . . . .	39
Opening a live-system dump with the crash tool . . . . .	40
<b>Chapter 9. Processing dumps</b> . . . . .	<b>43</b>
Reducing dump size . . . . .	43
Compressing a dump using makedumpfile . . . . .	44
Compressing a dump using gzip and split . . . . .	45
Preparing for analyzing a dump . . . . .	46
Sending a dump to IBM Support . . . . .	47
<b>Appendix A. Obtaining a dump with limited size</b> . . . . .	<b>49</b>
<b>Appendix B. Command summary</b> . . . . .	<b>51</b>
zgetdump - Copy and convert kernel dumps . . . . .	51
dumpconf - Configure panic or PSW restart action . . . . .	56
crash - Analyze kernel dumps . . . . .	59
vmconvert - Convert z/VM VMDUMPS for Linux . . . . .	59
vmur - Receive dumps from the z/VM reader . . . . .	60
<b>Accessibility</b> . . . . .	<b>61</b>
<b>Notices</b> . . . . .	<b>63</b>
Trademarks . . . . .	64
<b>Index</b> . . . . .	<b>65</b>



---

## Summary of changes

Summary of the changes made in the latest releases.

---

### Updates for Red Hat Enterprise Linux 7.3 and 7.4

This edition (SC34-2711-02) contains changes related to Red Hat Enterprise Linux 7.3 and 7.4

#### New information

- Information pertaining to preparing and planning for dumps is now collected in a new chapter, see Chapter 1, “Planning for dumps,” on page 1.
- Information about how to process a dump is now collected in a new chapter, see Chapter 9, “Processing dumps,” on page 43.

#### Changed Information

- Examples from the former “Appendix A. Examples for initiating dumps” have been integrated into their respective chapters.

This revision also includes maintenance and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

#### Deleted Information

- “Sharing dump devices” has been removed. The information is still available in older versions of this book.
- “Dumping NSSs” has been removed. The information is still available in older versions of this book.

---

### Updates for Red Hat Enterprise Linux 7.2

#### New information

- You can now create and analyze kernel dumps of Linux instances that run on an LPAR with multithreading enabled. See “Tools overview” on page 1.

#### Changed Information

- More details are given about files needed for dump analysis. See “Preparing for analyzing a dump” on page 46.

This revision also includes maintenance and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

#### Deleted Information

- None





---

## About this book

This book describes tools for obtaining dumps of Linux for IBM® System z® instances running Red Hat Enterprise Linux 7.4. This book describes how to use DASD, channel-attached tape, and SCSI disks, as well as how to use kdump and VMDUMP.

Unless stated otherwise, all z/VM® related information in this document assumes a current z/VM version, see [www.vm.ibm.com/techinfo/lpmigr/vmleos.html](http://www.vm.ibm.com/techinfo/lpmigr/vmleos.html).

As of July 2017, IBM z Systems™ is re-branded to IBM Z. In this document, *Linux on z Systems* refers to Linux running on LinuxONE or an IBM mainframe, including all IBM mainframe systems supported by Red Hat Enterprise Linux. In particular, this includes IBM z13™ (z13), IBM zEnterprise® BC12 (zBC12), IBM zEnterprise EC12 (zEC12), IBM zEnterprise 196 (z196), IBM zEnterprise 114 (z114).

For Red Hat Enterprise Linux product documentation, including what is new, known issues, and frequently asked questions, see the Red Hat Enterprise Linux documentation Web site at

[http://docs.redhat.com/docs/en-US/Red\\_Hat\\_Enterprise\\_Linux/](http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/)

You can find the latest version of this document on developerWorks® at

[www.ibm.com/developerworks/linux/linux390/documentation\\_red\\_hat.html](http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html)

### Authority

Most of the tasks described in this document require a user with root authority. In particular, writing to `procfs`, and writing to most of the described `sysfs` attributes requires root authority.

Throughout this document, it is assumed that you have root authority.

---

## Other relevant Linux on IBM z Systems publications

Another Linux on IBM z Systems publication for Red Hat Enterprise Linux 7 is available on developerWorks.

You can find the latest versions of this publication at [www.ibm.com/developerworks/linux/linux390/documentation\\_red\\_hat.html](http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html).

- *Device Drivers, Features, and Commands on Red Hat Enterprise Linux 7.4*, SC34-2710

For each of the following publications, the same web page points to the version that most closely reflects Red Hat Enterprise Linux 7.4:

- *How to Improve Performance with PAV*, SC33-8414
- *How to use FC-attached SCSI devices with Linux on z Systems*, SC33-8413
- *How to Set up a Terminal Server Environment on z/VM*, SC34-2596
- *Exploiting Enterprise PKCS #11 using openCryptoki*, SC34-2713
- *libica Programmer's Reference*, SC34-2602
- *Secure Key Solution with the Common Cryptographic Architecture Application Programmer's Guide*, SC33-8294

- *Linux on z Systems Troubleshooting, SC34-2612*

---

## Chapter 1. Planning for dumps

Be prepared before disaster strikes! Consider what dump method you want to use, what size dumps you need to handle, and what possibilities exist to limit the size or spread the dump over several devices.

---

### Tools overview

Different tools can be used for obtaining dumps for instances of Red Hat Enterprise Linux 7.4 running on IBM z Systems mainframes.

As of IBM z13, simultaneous multithreading is available for Linux in LPAR mode. Red Hat Enterprise Linux as of version 7.2 includes dump tools that can create dumps for both Linux instances with and without SMT enablement.

Dump tools from earlier versions of Red Hat Enterprise Linux are restricted to Linux instances without SMT enablement. Do not use dump disks that were prepared with stand-alone dump tools of earlier versions to create dumps of SMT-enabled Linux instances.

You can use the dump analysis tool **crash** to analyze a dump. Depending on your service contract, you might also want to send a dump to IBM support to be analyzed.

Table 1 summarizes the available dump tools:

Table 1. Dump tools summary

Dump aspect	kdump	DASD	Multi-volume DASD	SCSI	Tape	VMDUMP	Live-system dump with zgetdump
Environment	z/VM and LPAR	z/VM and LPAR	z/VM and LPAR	z/VM and LPAR	z/VM and LPAR	z/VM only	z/VM and LPAR
System size (see note 1 on page 2)	Large	Small	Large	Large	Large	Small	Large
Speed	Fast	Fast	Fast	Fast	Slow	Slow	Fast
Medium	Any available medium	ECKD™ or FBA (see note 2 on page 2) DASD	ECKD DASD	Linux file system	Tape cartridges	z/VM reader	Any available medium
Compression possible	While writing	No	No	While writing	Yes (see note 1 on page 2)	No	No
Dump filtering possible	While writing	When copying	When copying	When copying	When copying	When copying	No
Disruptive (see note 3 on page 2)	Yes	Yes	Yes	Yes	Yes	No	No
Stand-alone	No	Yes	Yes	Yes	Yes	No	No

**Note:**

1. For dump system sizes, see also “Maximum dump size by tool” on page 3.
2. SCSI disks can be emulated as FBA disks. This dump method can, therefore, be used for SCSI-only z/VM installations.
3. In this context, disruptive means that the dump process kills a running operating system.

### Note on device nodes

In all examples, the traditional device nodes for DASD, tape, and SCSI devices are used. You can also use the device nodes that **udev** creates for you.

## kdump

The **kdump** tool is made available through a Linux kernel and initial RAM disk that are preloaded in memory, along with a production system.

You do not have to install **kdump** on a dedicated dump device. The **kdump** system can access the memory that contains the dump of the production system through a **procfs** file.

Filtering out extraneous memory pages and compression can take place while the dump is written to persistent storage or transferred over a network. The smaller dump size can significantly reduce the write or transfer time, especially for large production systems.

Because **kdump** can write dumps through a network, existing file system facilities can be used to prevent multiple dumps from being written to the same storage space. Sharing space for dumps across an enterprise is possible.

## Stand-alone tools

Stand-alone tools are installed on a device on which you perform an IPL. Different tools are available depending on the device type.

Four stand-alone dump tools are shipped in the **s390utils** package as part of **zipl**:

- DASD dump tool for dumps on a single DASD device
- Multivolume DASD dump tool for dumps on a set of ECKD DASD devices
- Tape dump tool for dumps on (channel-attached) tape devices
- SCSI disk dump tool for dumps on SCSI disks

You need to install these tools on the *dump device*. A dump device is used to initiate a stand-alone dump by IPL-ing the device. It must have a stand-alone dump tool installed and should provide enough space for the dump. For Linux on z/VM, the dump device must be on subchannel set 0. For Linux in LPAR mode, the device can be on any subchannel set.

Typically, the system operator initiates a dump after a system crash, but you can initiate a dump at any time. To initiate a dump, you must IPL the dump device. This is destructive, that is, the running Linux operating system is killed. The IPL process writes the system memory to the IPL device (DASD and tape) or directly to a file on a SCSI disk.

You can configure a dump device that is automatically used when a kernel panic occurs. For more information, see “**dumpconf - Configure panic or PSW restart action**” on page 56.

For more information on **zipl**, refer to the **zipl** man page and to the **zipl** description in *Device Drivers, Features, and Commands on Red Hat Enterprise Linux 7.4*, SC34-2710. You can find the latest version of this document on developerWorks at:

[www.ibm.com/developerworks/linux/linux390/documentation\\_red\\_hat.html](http://www.ibm.com/developerworks/linux/linux390/documentation_red_hat.html)

## The VMDUMP tool

The **VMDUMP** tool is a part of z/VM and does not need to be installed separately.

Dumping with **VMDUMP** is not destructive. If you dump an operating Linux instance, the instance continues running after the dump is completed.

**VMDUMP** can also create dumps for z/VM guests that use z/VM named saved systems (NSS).

Do not use **VMDUMP** to dump large z/VM guests; the dump process is very slow. Dumping 1 GB of storage can take up to 15 minutes depending on the used storage server and z/VM version.

For more information on **VMDUMP** see *z/VM CP Commands and Utilities Reference*, SC24-6175.

## Live-system dump

You can create a kernel dump from a live system without disruption.

Use the **zgetdump** tool that is shipped with the s390utils-base RPM to create a kernel dump while the Linux system continues running. No dump device must be prepared, because the `/dev/crash` device node is used to create the dump.

---

## Maximum dump size by tool

The dump size depends on the size of the system for which the dump is to be created.

All dump methods require persistent storage space to hold the kernel and user space of this system.

### **kdump**

Initially uses the memory of the Linux instance for which a dump is to be created, and so supports any size. A persistent copy can be written to any medium of sufficient size. While writing, the dump size can be reduced through page filtering and compression.

### **DASD**

Depends on the disk size. For example, ECKD model A provides several hundreds of GB, depending on the storage server model. To write a large memory dump to single volume DASD, you can use:

- 3390 model 9 (up to 45 GB)
- 3390 model A (up to 1 TB)

### **Multivolume DASD**

Can be up to the combined size of 32 DASD partitions. To write a large memory dump to multiple DASD volumes, you can use up to 32 DASDs:

- 32 x 3390 model 9 (up to 1.4 TB)
- 32 x 3390 model A (up to 32 TB)

**SCSI** Depends on the capacity of the SCSI disk and which other data it contains. To write a large memory dump to SCSI disk, you can use:

**z/VM emulated FBA device that represents a real SCSI disk**

FBA disks can be defined with the CP command SET EDEVICE. These disks can be used as single-volume DASD dump disks. The SCSI disk size depends on your storage server setup.

**SCSI disk**

The SCSI disk size depends on your storage server setup. For SCSI dump partitions greater than 2 TB, you must use the GPT disk layout.

**Channel-attached tape**

Depends on the tape drive. For example, IBM TotalStorage Enterprise Tape System 3592 supports large dumps and also offers hardware compression. For large memory dumps, cartridges with up to 10 TB capacity are available.

**VMDUMP**

Depends on the available spool space. The slow dump speed can lead to very long dump times for large dumps. Although technically possible, the slow dump speed makes VMDUMP unsuitable for large dumps.

**zgetdump live-system dump**

The dump can be written to any medium of sufficient size.

See “Reducing dump size” on page 43 for information specific to large dumps.

---

## Dump methods compared

The process for preparing a dump device and obtaining a dump differs for the available dump methods.

Table 2. Comparing the dump methods.

Dump aspect	kdump	Stand-alone tools	VMDUMP	Live-system dump with zgetdump
Preparation	Reserve memory with the <code>crashkernel=</code> kernel parameter  Load the <code>kdump</code> kernel and the initial RAM disk into the memory of the production system Use <code>kexec</code> or <code>systemctl start kdump</code>	Write the stand-alone dump tool to the dump device ( <code>zipl</code> )  Define the panic shutdown action ( <code>dumpconf</code> )	Define the panic shutdown action ( <code>dumpconf</code> )	None
Dump trigger	<b>Automatic:</b> Kernel panic <b>Initiated by operator:</b> PSW restart	<b>Automatic:</b> Kernel panic <b>Initiated by operator:</b> IPL of the dump device	<b>Automatic:</b> Kernel panic <b>Initiated by operator:</b> z/VM CP <b>VMDUMP</b> command	<b>Initiated by operator:</b> <b>zgetdump</b> invocation
Initial dump space	Memory	Dump device	Spool device	Memory

Table 2. Comparing the dump methods (continued).

Dump aspect	kdump	Stand-alone tools	VMDUMP	Live-system dump with zgetdump
Accessing the initial dump	Through /proc/vmcore from the kdump instance (automatically done by kdump initrd)	Using <b>zgetdump</b> from a new Linux instance	Using <b>vmur -c</b> from a new Linux instance	Through /dev/crash
Copying the initial dump to the final dump store (and releasing the initial dump space)	Copied from the kdump instance to any available storage (automatically done by kdump initrd)	Copied from the new Linux instance to any available storage	Copied from the new Linux instance to any available storage	Copied from the current Linux instance to any available storage
Optional: Filtering the initial dump	Using /proc/vmcore and <b>makedumpfile</b> on the kdump instance (automatically done by kdump initrd)	Using <b>zgetdump</b> and <b>makedumpfile</b> on the new Linux instance	Using <b>zgetdump</b> and <b>makedumpfile</b> on the new Linux instance	Not recommended

## Automatic dumping

You can configure a dump device that is automatically used when a kernel panic occurs.

If you set up kdump, a kernel panic or PSW restart automatically triggers a dump. You can also use dumpconf to set up automatic dumping for standalone tools, or for your backup dump solution.

See “dumpconf - Configure panic or PSW restart action” on page 56 for how to set up dumpconf. Once you have set up the automation, you can cause a kernel panic to test the configuration.

## Testing automatic dump-on-panic

Cause a kernel panic to confirm that your dump configuration is set up to automatically create a dump if a kernel panic occurs.

### Before you begin

You need a Linux instance with active magic sysrequest functions.

### Procedure

Crash the kernel with a forced kernel panic.

If your method for triggering the magic sysrequest function is:	Enter:
A command on the 3270 terminal or line-mode terminal on the HMC	^~c
A command on the hvc0 terminal device	<b>Ctrl+O</b> c
Writing to procfs	echo c > /proc/sysrq-trigger

| **Note:** `Ctrl+o` means pressing `o` while holding down the control key.  
| See *Device Drivers, Features, and Commands on Red Hat Enterprise Linux 7.4*,  
| SC34-2710 for more details about the magic `sysrequest` functions.

## | **Results**

| The production system crashes. If `kdump` is set up correctly, the `kdump` kernel is  
| booted, the dump is created (the default directory is `/var/crash/`), and your  
| production system is rebooted.

---

## | **Sharing dump devices between different versions of Linux**

| Do not share dump devices between Linux installations with different major  
| releases.

| Always use the **`zgetdump`** command and the **`crash`** utility that are delivered with  
| your latest version of Red Hat Enterprise Linux. The latest **`crash`** utility can  
| process dumps that are created from Linux instances of the same or in earlier  
| versions of Red Hat Enterprise Linux.

| The latest **`zgetdump`** command can process dumps that are created with the `zipl`  
| (`s390utils` package) version in the same or in earlier versions of Red Hat Enterprise  
| Linux.

| As of Red Hat Enterprise Linux 7.2, `zipl` can prepare disks for dumps of  
| SMT-enabled Linux instances. You can use a dump disk that is prepared with an  
| SMT-enabled `zipl` version for both Linux instances with and without SMT  
| enablement.



---

## Chapter 2. Using kdump

You can use kdump to create system dumps for instances of Red Hat Enterprise Linux.

### Advantages of kdump

kdump offers these advantages over other dump methods:

- While writing the dump, you can filter out extraneous pages and compress the dump, and so handle large dumps in a short time.
- When writing dumps over a network, you can use existing file system facilities to share dump space without special preparations.

### Shortcomings of kdump

kdump has these drawbacks:

- kdump cannot be used for issues that occur before kdump is initialized, for example, for early boot problems. For such cases, use a standalone dump tool.
- kdump is not as reliable as the stand-alone dump tools. For critical systems, you can set up stand-alone dump tools as a backup, in addition to the kdump configuration (see “Failure recovery and backup tools” on page 9).
- For production systems that run in LPAR mode, kdump consumes memory (see “Memory consumption” on page 8).

---

## How kdump works on z Systems

You can set up kdump according to your needs.

With kdump, you do not need to install a dump tool on the storage device that is to hold a future dump. Instead, you use a kdump kernel, a Linux instance that controls the dump process.

The kdump kernel occupies a reserved memory area within the memory of the production system for which it is set up. The reserved memory area is defined with the `crashkernel=` kernel parameter. After the production system is started, the kdump init service loads the kdump kernel and its initial RAM disk (initrd) into the reserved memory area with the **kexec** tool.

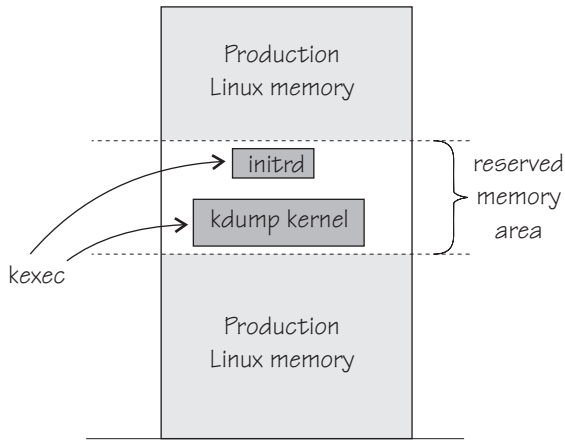


Figure 1. Running production system with preloaded kdump kernel and initial RAM disk

At the beginning of the dump process, the reserved memory area is exchanged with the lower memory regions of the crashed production system. The kdump system is then started and runs entirely in the memory that was exchanged with the reserved area. From the running kdump kernel, the memory of the crashed production system can be accessed as a virtual file, `/proc/vmcore`.

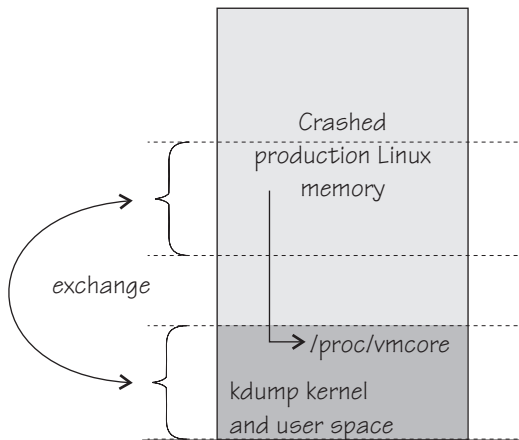


Figure 2. Running kdump kernel

This process is fast because the kdump kernel is started from memory, and no dump data needs to be copied up to this stage. For Red Hat Enterprise Linux, the **makedumpfile** tool in the kdump initrd writes a filtered and compressed version of the dump to a file on persistent storage, locally or over a network. Again, this method saves time because the dump is reduced in size while it is written or transferred.

By default, kdump initrd automatically IPLs the production system after the dump is written.

## Memory consumption

Although each Linux instance must be defined with additional memory for kdump, the total memory consumption for your z/VM installation does not increase considerably.

On most architectures, the inactive kdump system consumes the entire memory that is reserved with the `crashkernel=` kernel parameter.

For Linux on z/VM, only the kdump image and its initial RAM disk consume actual memory. The remaining reserved memory is withheld by the z/VM hypervisor until it is required in exchange for the lower memory region of the crashed production system.

Because the kdump image and initial RAM disk are not used during regular operations, z/VM swaps them out of memory some time after IPL. Thereafter, no real memory is occupied for kdump until it is booted to handle a dump.

For Linux in LPAR mode, the reserved memory area consumes real memory.

## Failure recovery and backup tools

If kdump fails, stand-alone dump tools or VMDUMP can be used as backup tools. Backup tools are, typically, set up only for vital production systems.

Because of being preloaded into memory, there is a small chance that parts of kdump are overwritten by malfunctioning kernel functions. The kdump kernel is, therefore, booted only if a checksum assures the integrity of the kdump kernel and initial RAM disk. This failure can be recovered automatically by setting up a backup dump tool with the **dumpconf** service or through a backup dump that is initiated by a user. See “`dumpconf` - Configure panic or PSW restart action” on page 56.

A second possible failure is the kdump system itself crashing during the dump process. This failure occurs, for example, if the reserved memory area is too small for the kdump kernel and user space. For this failure, initiate a backup dump, which captures data for both the crashed production system and the crashed kdump kernel. You can separate this data with the **zgetdump --select** option. See “`zgetdump` - Copy and convert kernel dumps” on page 51.

---

## Setting up kdump

Red Hat Enterprise Linux provides several ways of setting up kdump.

### About this task

You can choose between the following methods of setting up kdump:

- The Kernel Dump Configuration utility **system-config-kdump**: Graphical tool with more configuration options. For a configuration example, see the chapter about configuring kdump in the graphical user interface in the *Red Hat Enterprise Linux 7.4 Kernel Crash Dump Guide*.
- Manually using configuration files such as `/etc/kdump.conf`: For a configuration example, see the chapter on kdump in the *Red Hat Enterprise Linux 7.4 Kernel Crash Dump Guide*.
- The `crashkernel` boot parameter: For systems with more than 4 GB of memory the configuration `crashkernel=auto` enables kdump by default. For smaller systems you must explicitly define the `crashkernel` parameter in `/etc/zipl.conf`, for example: `crashkernel=192M`. For details, see the chapter on kdump in the *Red Hat Enterprise Linux 7.4 Kernel Crash Dump Guide*.

## What to do next

As a backup, you can set up a stand-alone dump tool in addition to kdump. See “dumpconf - Configure panic or PSW restart action” on page 56 about how to run a backup tool automatically, if kdump fails.

---

## Initiating a dump

A kernel panic automatically triggers the dump process with kdump. When your Linux system does not respond and kdump is not triggered automatically, depending on your system environment, there are additional methods for triggering the dump process.

### About this task

With kdump installed, a kernel panic or PSW restart trigger kdump rather than the shutdown actions defined in `/sys/firmware`. The definitions in `/sys/firmware` are used only if an integrity check for kdump fails (see also “Failure recovery and backup tools” on page 9 and “dumpconf - Configure panic or PSW restart action” on page 56).

To trigger kdump, use one of the methods according to your environment:

- For Linux on z/VM: Run the z/VM CP **system restart** command. See “z/VM guest example” for an example .
- For Linux in LPAR mode: Run the **PSW restart** task on the HMC. See “LPAR example” on page 11 for an example.

**Tip:** On both z/VM and LPAR you can use the `diag288` watchdog to trigger kdump. The default setup triggers the correct actions. For more details about the `diag288` watchdog, see *Device Drivers, Features, and Commands on Red Hat Enterprise Linux 7.4*, SC34-2710

### Results

After kdump is triggered, first kernel messages from the booting kdump kernel and later dump progress messages are issued. The messages are written to the Operating System Messages applet of the HMC for LPAR and to the 3270 terminal for z/VM. The kdump scripts copy the dump and reboot automatically.

### What to do next

Verify that your production system is up and running again. Send the created dump to a service organization.

## z/VM guest example

With kdump, you do not need a dump device to initiate the dump.

### Before you begin

Your Linux instance must be set up for kdump as described in “Setting up kdump” on page 9.

## Procedure

Issue the **system restart** z/VM CP command. For example from a 3270 terminal emulation for the Linux instance to be dumped:

```
#cp system restart
```

Boot messages for the kdump kernel indicate that the dump process started.

## LPAR example

You can initiate a kdump process on an LPAR from an HMC (Hardware Management Console) or SE (Support Element).

### About this task

The following description refers to an HMC, but the steps also apply to an SE.

### Procedure

1. In the navigation pane of the HMC, expand Systems Management and Servers and select the mainframe system you want to work with. A table of LPARs is displayed in the content area.
2. Select the LPAR for which you want to initiate the dump.
3. In the **Tasks** area, click **PSW restart**, which is located in the **Recovery** section. This initiates the dump process. Figure 3 shows an example of an HMC with a selected mainframe system and LPAR. The **PSW restart** task can be seen in the **Tasks** area.

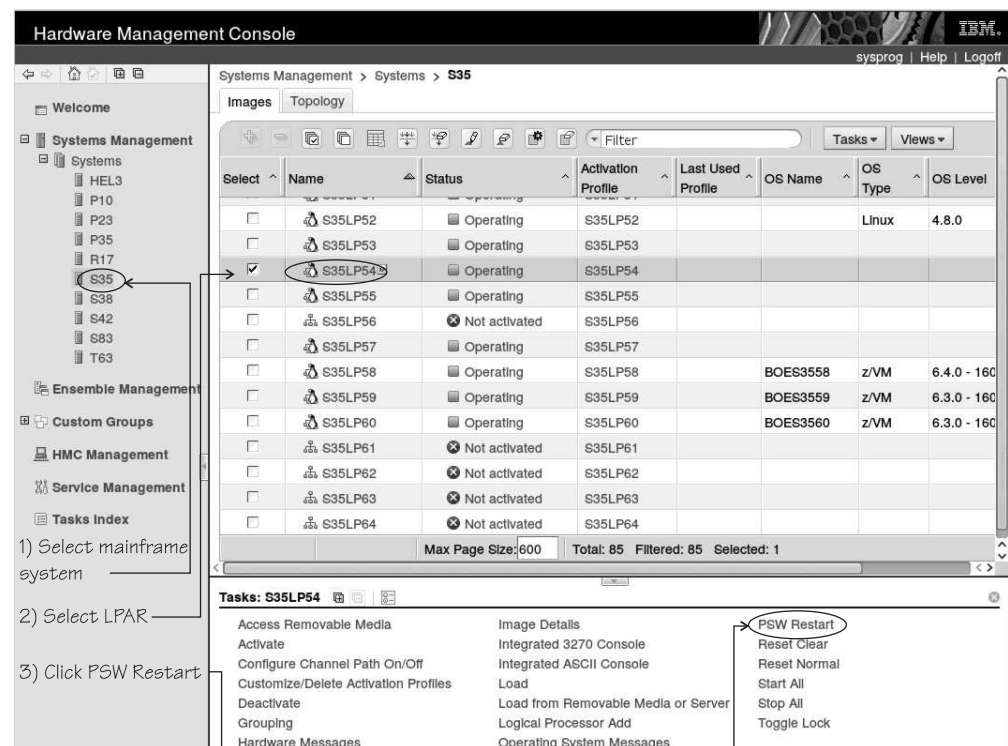


Figure 3. HMC with the PSW restart task

4. Wait until the dump process completes. Click the **Operating System Messages** icon for progress and error information.

## **Results**

|

kdump automatically collects the dump and reboots Linux.

---

## Chapter 3. Using a DASD dump device

To use a DASD dump device, you need to install the stand-alone DASD dump tool and perform the dump process. Then, copy the dump to a file in a Linux file system.

DASD dumps are written directly to a DASD partition that was not formatted with a file system. The following DASD types are supported:

- ECKD DASDs
  - 3380
  - 3390
- FBA DASDs

---

### Installing the DASD dump tool

Install the DASD dump tool on an unused DASD partition. Memory dumps are written to this partition.

#### Before you begin

You need an unused DASD partition with enough space (memory size + 10 MB) to hold the system memory. If the system memory exceeds the capacity of a single DASD partition, use the multivolume dump tool, see Chapter 4, “Using DASD devices for multi-volume dump,” on page 19.

#### About this task

The examples assume that `/dev/dasdc` is the dump device and that you want to dump to the first partition `/dev/dasdc1`.

The steps that you need to perform for installing the DASD dump tool depend on your type of DASD, ECKD or FBA:

- If you are using a DASD of type ECKD, perform all three of the following steps.
- If you are using a DASD of type FBA, skip steps 1 and 2 and perform step 3 only.

#### Procedure

1. (ECKD only) Format your DASD with **dasdfmt**. Use a block size of 4 KB. For example:

```
# dasdfmt -f /dev/dasdc -b 4096
```

2. (ECKD only) Create a partition with **fdasd**. The partition must be sufficiently large (the memory size + 10 MB). For example:

```
# fdasd /dev/dasdc
```

3. Install the dump tool with the **zipl** command. Specify the dump device on the command line. For example:

```
# zipl -d /dev/dasdc1
```

**Note:** When you use a DASD of type ECKD that is formatted with the traditional Linux disk layout `ldl`, the dump tool must be reinstalled with `zip1` after each dump.

---

## Initiating a DASD dump

You can initiate a dump from a DASD device.

### Procedure

To obtain a dump with the DASD dump tool, perform the following main steps:

1. Stop all CPUs.
2. Store status on the IPL CPU.
3. IPL the dump tool on the IPL CPU.

**Note:** Do not clear storage!

The dump process can take several minutes depending on the device type you are using and the amount of system memory. After the dump completes, the IPL CPU should go into disabled wait.

The following PSW indicates that the dump process completed successfully:

```
(64-bit) PSW: 00020000 80000000 00000000 00000000
```

Any other disabled wait PSW indicates an error.

After the dump tool is IPLed, messages that indicate the progress of the dump are written to the console:

```
Dumping 64 bit OS
00000032 / 00000256 MB
00000064 / 00000256 MB
00000096 / 00000256 MB
00000128 / 00000256 MB
00000160 / 00000256 MB
00000192 / 00000256 MB
00000224 / 00000256 MB
00000256 / 00000256 MB
Dump successful
```

### Results

You can IPL Linux again.

## z/VM guest example

In this example, a dump to DASD device 193 is initiated from z/VM.

### Example

If 193 is the dump device:

```
#cp cpu all stop
#cp store status
#cp i 193
```

On z/VM, a three-processor machine in this example, you will see messages about the disabled wait:



```

01: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
02: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.
"CP entered; disabled wait PSW 00020000 80000000 00000000 00000000"

```

You can now IPL your Linux instance and resume operations.

## Dumping an LPAR using the HMC or SE web interface example

In this example, an LPAR dump to a DASD device is initiated from an HMC (Hardware Management Console) or SE (Support Element).

### About this task

The following description refers to an HMC, but the steps also apply to an SE.

### Procedure

1. In the navigation pane of the HMC, expand Systems Management and Servers and select the mainframe system you want to work with. A table of LPARs is displayed in the content area.
2. Select the LPAR for which you want to initiate the dump.
3. In the **Tasks** area, click **Stop all** (which is located in the **Recovery** section) to stop all CPUs. Confirm when you are prompted to do so.
4. In the **Tasks** area, click **Load** to display the Load panel. Figure 4 shows an example of an HMC with a selected mainframe system and LPAR. The **Load** and **Stop all** tasks can be seen in the Tasks area.

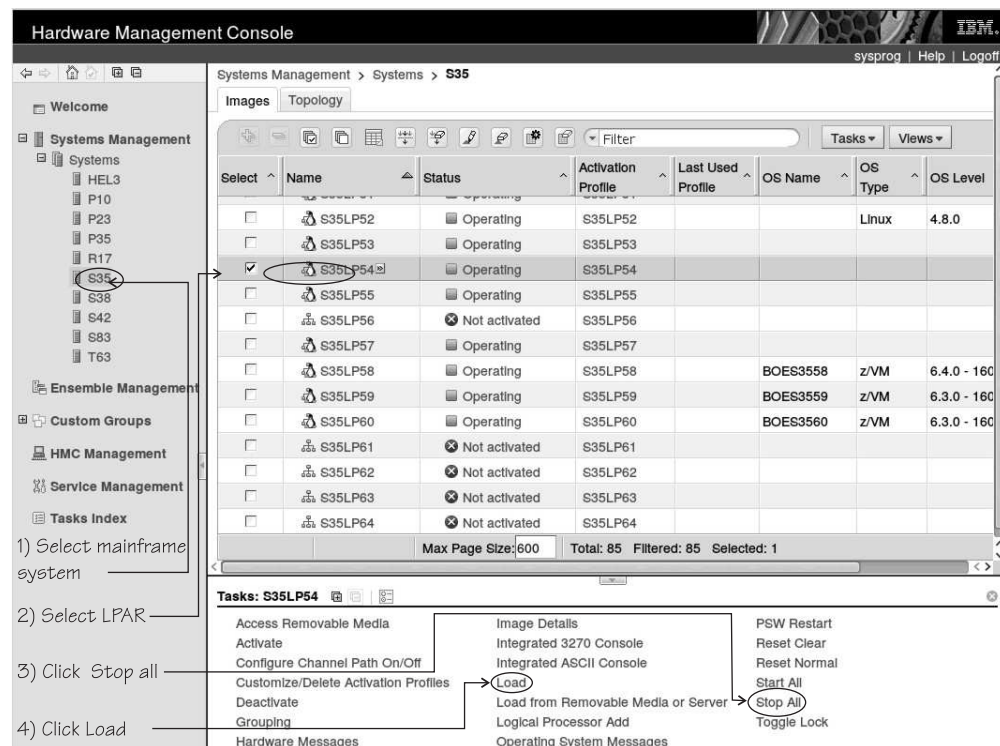


Figure 4. HMC with the Load and Stop all tasks

5. Select Load type "Normal".

6. Select the **Store status** check box.
7. Type the device number of the dump device into the **Load address** field. To IPL from a subchannel set other than 0, specify five digits: The subchannel set ID followed by the device number, for example 1E711. Figure 5 shows a Load panel with all entries and selections required to start the dump process for a DASD or tape dump device.

Figure 5. Load panel for dumping to DASD

8. Click **OK** to start the dump process.
9. Wait until the dump process completes. Click the **Operating System Messages** icon for progress and error information.

## Results

When the dump has completed successfully, you can IPL Linux again.

## Automatic dump example

On both z/VM and LPAR, you can use the dumpconf service to set up automatic dumping. In this example, a dump is automatically triggered when a kernel panic occurs.

### About this task

Use the dumpconf service to set up automatic dumping. To set up dumping, edit the configuration file `/etc/sysconfig/dumpconf`.

### Example

Example configuration for a CCW dump device (DASD):

```
ON_PANIC=dump
DUMP_TYPE=ccw
DEVICE=0.0.4714
```

For details on how to set up `dumpconf`, see “`dumpconf` - Configure panic or PSW restart action” on page 56.

---

## Copying the dump from DASD with `zgetdump`

You can copy a DASD dump to a file system by using the `zgetdump` tool.

### About this task

By default, the `zgetdump` tool takes the dump device as input and writes its contents to standard output. To write the dump to a file system, you must redirect the output to a file.

### Procedure

Assuming that the dump is on DASD device `/dev/dasdc1` and you want to copy it to a file named `dump_file`:

```
# zgetdump /dev/dasdc1 > dump_file
```

### What to do next

You can use `zgetdump` to display information about the dump. See “Checking whether a DASD dump is valid and printing the dump header” on page 54 for an example.

For general information about `zgetdump`, see “`zgetdump` - Copy and convert kernel dumps” on page 51 or the man page.



---

## Chapter 4. Using DASD devices for multi-volume dump

You can handle large dumps, up to the combined size of 32 DASD partitions, by creating dumps across multiple volumes.

### Before you begin

Prepare a set of ECKD DASD devices for a multivolume dump, install the stand-alone dump tool on each DASD device that is involved. Then, perform the dump process, and copy the dump to a file in a Linux file system.

### About this task

You can specify up to 32 partitions on ECKD DASD volumes for a multivolume dump. The dump tool is installed on each volume involved. The volumes must be:

- In subchannel set 0.
- Formatted with the compatible disk layout (cdl, the default option when you use the **dasdfmt** command.)

You must specify block size 4096 for **dasdfmt**.

For example, Figure 6 shows three DASD volumes, *dasdb*, *dasdc*, and *dasdd*, with four partitions that are selected to contain the dump. To earmark the partition for dump, a dump signature is written to each partition.

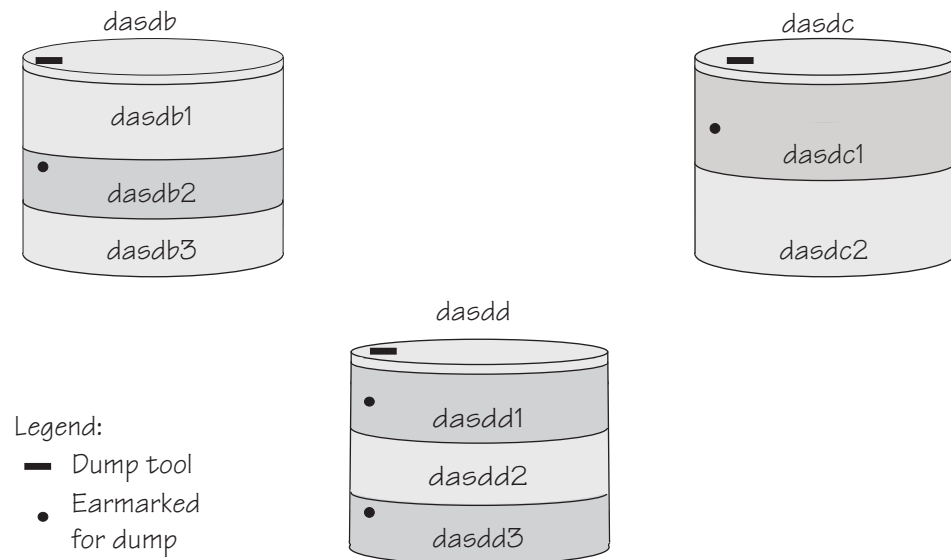


Figure 6. Three DASD volumes with four partitions for a multivolume dump

The partitions need to be listed in a configuration file, for example:

```
/dev/dasdb2  
/dev/dasdc1  
/dev/dasdd1  
/dev/dasdd3
```

You can define a maximum of three partitions on one DASD. All three volumes are prepared for IPL; regardless of which you use the result is the same.

The following sections take you through the entire process of creating a multivolume dump.

---

## Installing the multi-volume DASD dump tool

This example shows how to perform the dump process on two partitions, /dev/dasdc1 and /dev/dasdd1, which reside on ECKD volumes /dev/dasdc and /dev/dasdd.

### About this task

Assume that the corresponding device bus-IDs (as displayed by **lsdasd**) are 0.0.4711 and 0.1.4712.

### Procedure

1. Format both dump volumes with **dasdfmt**. The command shown uses the default cdl (compatible disk layout) and specifies a block size of 4KB.

```
# dasdfmt -f /dev/dasdc -b 4096
# dasdfmt -f /dev/dasdd -b 4096
```

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

```
# fdasd /dev/dasdc
# fdasd /dev/dasdd
```

3. Create a file named `mvdump.conf` containing the device nodes of the two partitions, separated by one or more line feed characters (0x0a). The file's contents are as follows:

```
/dev/dasdc1
/dev/dasdd1
```

4. Prepare the volumes using the **zip1** command. Specify the dump list on the command line:

```
# zip1 -M mvdump.conf
Dump target: 2 partitions with a total size of 1234 MB.
Warning: All information on the following partitions will be lost!
/dev/dasdc1
/dev/dasdd1
Do you want to continue creating multi-volume dump partitions (y/n)?
```

### Results

Now the two volumes /dev/dasdc and /dev/dasdd with device bus-IDs 0.0.4711 and 0.1.4712 are prepared for a multivolume dump. Use the **-device** option of **zgetdump** to display information about these volumes:

```
# zgetdump -d /dev/dasdc
Dump device info:
Dump tool.....: Multi-volume DASD dump tool
Version.....: 2
Architecture.....: s390x (64 bit)
Dump size limit...: none
Force specified...: no

Volume 0: 0.0.4711 (online/valid)
Volume 1: 0.1.4712 (online/valid)
```

During **zipl** processing both partitions were earmarked for dump with a valid dump signature. The dump signature ceases to be valid when data other than dump data is written to the partition. For example, writing a file system to the partition overwrites the dump signature. Before writing memory to a partition, the dump tool checks the partition's signature and exits if the signature is invalid. Thus any data inadvertently written to the partition is protected.

You can circumvent this protection, for example, if you want to use a swap space partition for dumping, by using the **zipl** command with the **--force** option. This option inhibits the dump signature check, and any data on the device is overwritten. Exercise great caution when using the force option.

The **zipl** command also takes a size specification, see Appendix A, “Obtaining a dump with limited size,” on page 49. For more details on the **zipl** command, see *Device Drivers, Features, and Commands on Red Hat Enterprise Linux 7.4*, SC34-2710.

---

## Initiating a multi-volume DASD dump

After preparing the DASD volumes, you can initiate a multi-volume dump by performing an IPL from one of the prepared volumes.

### Procedure

To obtain a dump with the multivolume DASD dump tool, perform the following steps:

1. Stop all CPUs.
2. Store status on the IPL CPU.
3. IPL the dump tool using one of the prepared volumes, either 0.0.4711 or 0.1.4712.

**Note:** Do not clear storage!

For an example performed for a single DASD on the HMC, see “Dumping an LPAR using the HMC or SE web interface example” on page 15. For an example performed for a single DASD on z/VM, see “z/VM guest example” on page 14.

The dump process can take several minutes depending on each volume's block size and the amount of system memory. After the dump has completed, the IPL CPU should go into disabled wait.

The following PSW indicates that the dump process has completed successfully:  
(64-bit) PSW: 00020000 80000000 00000000 00000000

Any other disabled wait PSW indicates an error.

After the dump tool is IPLed, messages that indicate the progress of the dump are written to the console:

```
Dumping 64 bit OS
Dumping to: 0.0.4711
00000128 / 00001024 MB
00000256 / 00001024 MB
00000384 / 00001024 MB
00000512 / 00001024 MB
Dumping to: 0.1.4712
00000640 / 00001024 MB
00000768 / 00001024 MB
00000896 / 00001024 MB
00001024 / 00001024 MB
Dump successful
```

## Results

You can IPL Linux again.

---

## Copying a multi-volume dump to a file

Use the **zgetdump** command to copy the multi-volume dump.

### About this task

This example assumes that the two volumes `/dev/dasdc` and `/dev/dasdd` (with device bus-IDs 0.0.4711 and 0.1.4712) contain the dump. Dump data is spread along partitions `/dev/dasdc1` and `/dev/dasdd1`.

### Procedure

Use **zgetdump** without any options to copy the dump parts to a file:

```
# zgetdump /dev/dasdc > multi_volume_dump_file
Format Info:
Source: s390mv
Target: s390
Copying dump:
00000000 / 00001024 MB
00000171 / 00001024 MB
00000341 / 00001024 MB
00000512 / 00001024 MB
00000683 / 00001024 MB
00000853 / 00001024 MB
00001024 / 00001024 MB
Success: Dump has been copied
```

If you want to check only the validity of the multivolume dump rather than copying it to a file, use the `-info` option with **zgetdump**. See “Checking whether a DASD dump is valid and printing the dump header” on page 54 for an example.



---

## Chapter 5. Using a tape dump device

You can use a channel-attached tape as a dump device. To use a tape, you need to install the stand-alone tape dump tool and perform the dump process. Then, copy the dump to a file in a Linux file system.

The following tape devices are supported:

- 3480
- 3490
- 3590
- 3592

The following sections take you through the entire process of creating a dump on a tape device.

---

### Installing the tape dump tool

Install the tape dump tool on the tape that is to hold the dump.

#### Before you begin

Have enough empty tapes ready to hold the system memory (memory size + 10 MB).

#### About this task

The examples assume that `/dev/ntibm0` is the tape device that you want to dump to.

#### Procedure

Perform these steps to install the tape dump tool:

1. Insert an empty dump cartridge into your tape device.
2. Ensure that the tape is rewound.
3. Install the dump tool with the `zip1` command. Specify the dump device on the command line. For example:

```
# zip1 -d /dev/ntibm0
```

---

### Initiating a tape dump

Initiate a tape dump by performing an IPL on the IPL CPU.

#### Procedure

To obtain a dump with the tape dump tool, perform the following main steps:

1. Ensure that the tape is rewound.
2. Stop all CPUs.
3. Store status on the IPL CPU.
4. IPL the dump tool on the IPL CPU.

**Note:** Do not clear storage!

The dump tool writes the number of dumped MB to the tape drive message display.

The dump process can take several minutes, depending on the device type you are using and the amount of system memory available. When the dump is complete, the message `dump*end` is displayed and the IPL CPU goes into disabled wait.

The following PSW indicates that the dump was taken successfully:

(64-bit) PSW: 00020000 80000000 00000000 00000000

Any other disabled wait PSW indicates an error.

After the dump tool is IPLed, messages that indicate the progress of the dump are written to the console:

```
Dumping 64 bit OS
00000032 / 00000256 MB
00000064 / 00000256 MB
00000096 / 00000256 MB
00000128 / 00000256 MB
00000160 / 00000256 MB
00000192 / 00000256 MB
00000224 / 00000256 MB
00000256 / 00000256 MB
Dump successful
```

Messages might be shown on the tape display.

*number*

The number of MB dumped.

`dump*end`

The dump process ended successfully.

5. You can IPL Linux again.

## z/VM guest example

You can initiate a dump to tape from a Linux instance that is running as a z/VM guest.

### Procedure

If 193 is the tape device:

1. Rewind the tape:

```
#cp rewind 193
```

2. Stop all CPUs:

```
#cp cpu all stop
```

3. Store status:

```
#cp store status
```

4. IPL the tape device:

```
#cp i 193
```

## Results

On z/VM, a three-processor machine in this example, you will see messages about the disabled wait:

```
01: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.  
02: The virtual machine is placed in CP mode due to a SIGP stop from CPU 00.  
"CP entered; disabled wait PSW 00020000 80000000 00000000 00000000"
```

You can now IPL your Linux instance and resume operations.

## LPAR example

You can initiate a dump to tape on an LPAR from an HMC (Hardware Management Console) or SE (Support Element).

### About this task

The following description refers to an HMC, but the steps also apply to an SE.

### Procedure

1. In the navigation pane of the HMC, expand Systems Management and Servers and select the mainframe system you want to work with. A table of LPARs is displayed in the content area.
2. Select the LPAR for which you want to initiate the dump.
3. In the **Tasks** area, click **Stop all** to stop all CPUs. Confirm when you are prompted to do so.
4. In the **Tasks** area, click **Load** to display the Load panel. Figure 7 on page 26 shows an example of an HMC with a selected mainframe system and LPAR. The **Load** and **Stop all** tasks can be seen in the **Tasks** area.

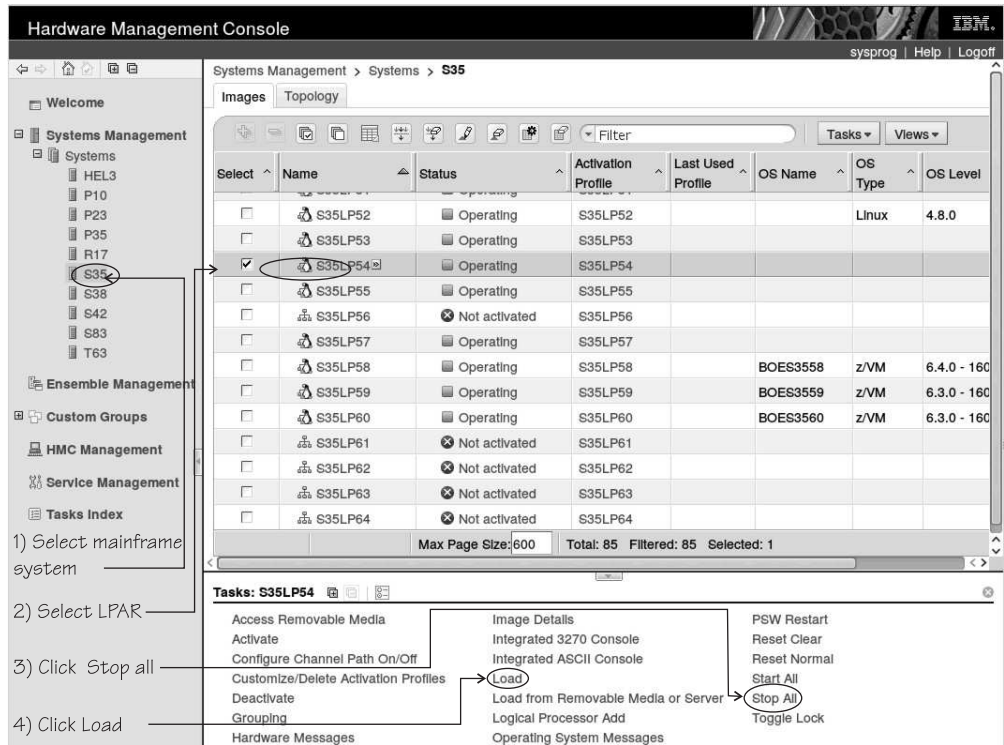


Figure 7. HMC with the **Load** and **Stop all** tasks

5. Select **Load** type "Normal".
6. Select the **Store status** check box.
7. Type the device number of the dump device into the Load address field. To IPL from a subchannel set other than 0, specify five digits: The subchannel set ID followed by the device number, for example 1E711. Figure 8 shows a Load panel with all entries and selections required to start the dump process for a channel-attached tape dump device.

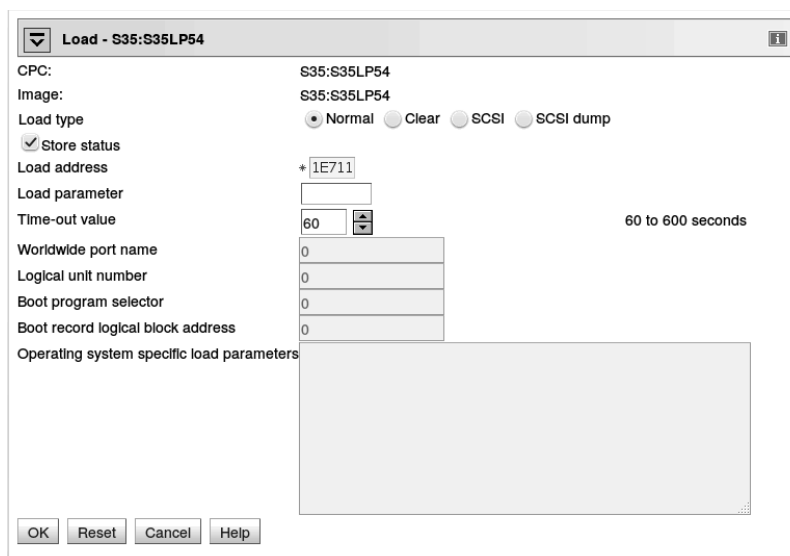


Figure 8. Load panel for dumping to tape

8. Click **OK** to start the dump process.
9. Wait until the dump process completes. Click the **Operating System Messages** icon for progress and error information.

## Results

When the dump has completed successfully, you can IPL Linux again.

---

## Copying the dump from tape

You can copy a tape dump to a file system with the **zgetdump** tool.

### Before you begin

The **mt** utility must be installed.

## Preparing the dump tape

You need to rewind the tape, and find the correct position on the tape to start copying from.

### About this task

Use the **mt** tool to manipulate the tape.

### Procedure

1. Rewind the tape.

For example:

```
# mt -f /dev/ntibm0 rewind
```

2. Skip the first file on the tape (this file is the dump tool itself).

For example:

```
# mt -f /dev/ntibm0 fsf
```

## Using the zgetdump tool to copy the dump

Use the **zgetdump** tool to copy the dump file from the tape to a file system.

### Before you begin

The tape must be in the correct position (see “Preparing the dump tape”).

### About this task

By default, the **zgetdump** tool takes the dump device as input and writes its contents to standard output. To write the memory dump to a file system, you must redirect the output to a file.

The example assumes that the dump is on tape device `/dev/ntibm0`.

## Procedure

Copy the dump from tape to a file named `dump_file` in the file system:

```
# zgetdump /dev/ntibm0 > dump_file
```

For general information on **zgetdump**, see “zgetdump - Copy and convert kernel dumps” on page 51 or the man page.

## Checking whether a dump is valid, and printing the dump header

To check whether a dump is valid, use the **zgetdump** command with the **-i** option.

### Procedure

1. Ensure that the volume is loaded.
2. Skip the first file on the tape (this file is the dump tool itself):

```
# mt -f /dev/ntibm0 fsf
```

3. Issue the **zgetdump** command with the **-i** option:

```
# zgetdump -i /dev/ntibm0
```

The **zgetdump** command goes through the dump until it reaches the end. See also “Using zgetdump to copy a tape dump” on page 53

---

## Chapter 6. Using a SCSI disk dump device

You can use SCSI disks that are accessed through the `zfcpx` device driver as dump devices. SCSI disk dumps are written as files in an existing file system on the dump partition. No copying is necessary.

---

### Installing the SCSI disk dump tool

You install the SCSI dump tool with the `zipl` command.

#### Before you begin

- The kernel-kdump RPM (named `kernel-kdump-<version>.s390x.rpm`) must be installed on your system.
- The dump directory needs enough free space (memory size + 10 MB) to hold the system memory.

#### About this task

The SCSI dump tool (also referred to as the SCSI Linux System Dumper, or SD) is written to one partition, referred to here as the *target partition*. The memory dump can be written to a second partition, the *dump partition*, provided it is on the same physical disk. Only the target partition needs to be mounted when `zipl` is run. In a single-partition configuration, the target partition is also the dump partition.

### SCSI dump tool parameters

When you install the SCSI disk dump tool, the following parameters can be specified in a 'parameters' line in the `zipl` configuration file or specified in the `zipl` command line with the `-P` option

#### Parameters

`dump_dir=<directory>`

Path to the directory (relative to the root of the dump partition) to which the dump file is to be written. This directory is specified with a leading slash. The directory must exist when the dump is initiated.

For example, if the dump partition is mounted as `/dumps`, and the parameter `dump_dir=/mydumps` is defined, the dump directory would be accessed as `/dumps/mydumps`.

The default is `/` (the root directory of the partition).

`dump_mode=interactive | auto`

Action that is taken if there is no room on the file system for the new dump file. **interactive** prompts the user to confirm that the memory dump with the lowest number is to be deleted. **auto** automatically deletes this file.

The default is **interactive**.

In rare cases, you might want to complement or overwrite the SCSI dump tool parameters that were configured with `zipl`. For example, you might want to change the dump mode setting when you initiate the dump. How you specify such parameters depends on whether your Linux instance runs in LPAR mode or as a z/VM guest. For more information, see the SCSI examples in Examples for initiating dumps.

## Combined dump and target partition

A single partition on a SCSI device can be used as both the dump partition and target partition.

### About this task

This example assumes that `/dev/sda` is a SCSI device that contains no data and is to be used exclusively as a dump device. Because no other data is to be stored on the device, a single partition is created that serves as both dump and target partition.

The example further assumes that `/dev/sda` is IPLed with the following parameters:

- devno: 0.0.4711
- wwpn: 0x4712076300ce93a7
- lun: 0x4712000000000000

### Procedure

1. Ensure that the SCSI disk is not part of a multipath device. Use the `lsblk` command to check the disk, for example:

```
# lsblk /dev/sda
NAME                                MAJ:MIN RM  SIZE RO  TYPE  MOUNTPOINT
sda                                  8:0      0   20G  0  disk
├─36005076307ffc5e30000000000002a4 252:3    0   20G  0  mpath
│  └─36005076307ffc5e30000000000002a4p1 252:4    0   256M  0  part
│     └─36005076307ffc5e30000000000002a4p2 252:7    0    2G  0  part
│        └─36005076307ffc5e30000000000002a4p3 252:9    0  17.8G  0  part
```

The row with `TYPE=mpath` shows the name of the multipath device of which `/dev/sda` is part. In this example, the device is `36005076307ffc5e30000000000002a4`. If there is no device with type `mpath` in the tree, then `/dev/sda` is not part of a multipath device in the current configuration. If there is a multipath device name, issue the following command to temporarily remove it:

```
# multipath -f 36005076307ffc5e30000000000002a4
```

The command releases the individual paths to the SCSI disk so that the disk can be used in the following steps.

2. Create a single partition with `fdisk`, using the PC-BIOS layout:

For example:

```
# fdisk /dev/sda
```

The created partition is `/dev/sda1`.

3. Format this partition with the `xfs` or `ext4` file system.

For example:

```
# mkfs.ext4 /dev/sda1
```

4. Mount the partition at a mount point of your choice and create a subdirectory to hold the dump files.



For example:

```
# mount /dev/sda1 /dumps
# mkdir /dumps/mydumps
```

5. Install the dump tool with the **zip1** command. Specify the dump device on the command line.

For example:

```
# zip1 -D /dev/sda1 -t /dumps -P "dump_dir=/mydumps"
```

6. Unmount the file system:

```
# umount /dumps
```

## Results

When you IPL `/dev/sda` (specified by the WWPN, LUN, and device-bus ID) using boot program selector 1 or 0 (default), the dump is written to directory `mydumps` on partition 1 of `/dev/sda`. The boot program selector is located on the load panel, see Figure 10 on page 35 for an example.

---

## Initiating a SCSI dump

To initiate the dump, IPL the SCSI dump tool with the SCSI dump load type. To IPL the dump tool, specify its WWPN, LUN, and device-bus ID.

### About this task

The dump process can take several minutes depending on the device type you are using and the amount of system memory. The dump progress and any error messages are reported on the operating system messages console.

### Procedure

IPL the SCSI dump tool. After the dump tool is IPLed, messages that indicate the progress of the dump are written to the console:

```
Uncompressing Linux...
Ok, booting the kernel.

[ 0.076456] zdump: DETECTED 'S390X (64 bit) OS'
Linux System Dumper starting
Version 2.2 (64 bit)
Linux version 3.10.0-685.el7.s390x.kdump (mockbuild@s390-005.build.eng.bos.redhat.com)
(gcc version 4.8.5 20150623 (Red Hat 4.8.

DUMP PARAMETERS:
=====
devno   : 0.0.4711
wwpn    : 0x4712076300ce93a7
lun     : 0x4712000000000000
conf    : 0
partition: 1
directory: /mydumps
compress : none

MOUNT DUMP PARTITION:
=====
DONE.
```

```
DUMP PROCESS STARTED:
=====
dump file: dump.0
  0 MB of 2048 MB ( 0.0% )
 1739 MB of 2048 MB ( 84.9% )
 2048 MB of 2048 MB (100.0% )

DUMP 'dump.0' COMPLETE
```

## Results

The dump process creates a new dump file in the dump directory. All dumps are named `dump.<n>`, where `<n>` is the dump number. A new dump receives the next highest dump number out of all dumps in the dump directory (see the `dump_dir` parameter under “SCSI dump tool parameters” on page 29).

For example, if there are already two dump files named `dump.0` and `dump.1` in the dump directory, the new dump is named `dump.2`.

When the dump completes successfully, you can IPL Linux again.

You do not need to convert the dump or copy it to a different medium. To access the dumps, mount the dump partition.

## z/VM guest example

You can initiate a dump to a SCSI disk using z/VM.

### Before you begin

SCSI dump from z/VM is supported as of z/VM 5.4.

### About this task

Assume your SCSI dump disk has the following parameters:

- WWPN: 4712076300ce93a7
- LUN: 4712000000000000
- FCP adapter device number: 4711

## Results

Messages on the operating system console will show when the dump process is finished.

### Example

```
#cp set dumpdev portname 47120763 00ce93a7 lun 47120000 00000000
#cp ipl 4711 dump
```

### What to do next

You can now IPL your Linux instance and resume operations.

In rare cases, you might want to overwrite or complement the existing SCSI dump tools parameters that have been configured with `zipl`. For example, you might

want to change the dump mode setting. You can use a command of this form to specify SCSI dump tools parameters to be concatenated to the existing parameters:

```
#cp set dumpdev scpdata '<parameters>'
```

Enter this command before entering the **IPL** command.

In contrast to SCSI IPL configurations, where you can use a leading equal sign to replace all kernel parameters, you cannot use a leading equal sign to replace all SCSI dump tool parameters. Specifying the parameters with a leading equal sign causes the dump to fail.

## LPAR example

You can initiate a dump process on an LPAR from an HMC (Hardware Management Console) or SE (Support Element).

### About this task

The following description refers to an HMC, but the steps also apply to an SE.

### Procedure

1. In the navigation pane of the HMC, expand Systems Management and Servers and select the mainframe system you want to work with. A table of LPARs is displayed in the content area.
2. Select the LPAR for which you want to initiate the dump.
3. In the **Tasks** area, click **Load** to display the Load panel. Figure 9 on page 34 shows an example of an HMC with a selected mainframe system and LPAR. The **Load** task can be seen in the **Tasks** area.

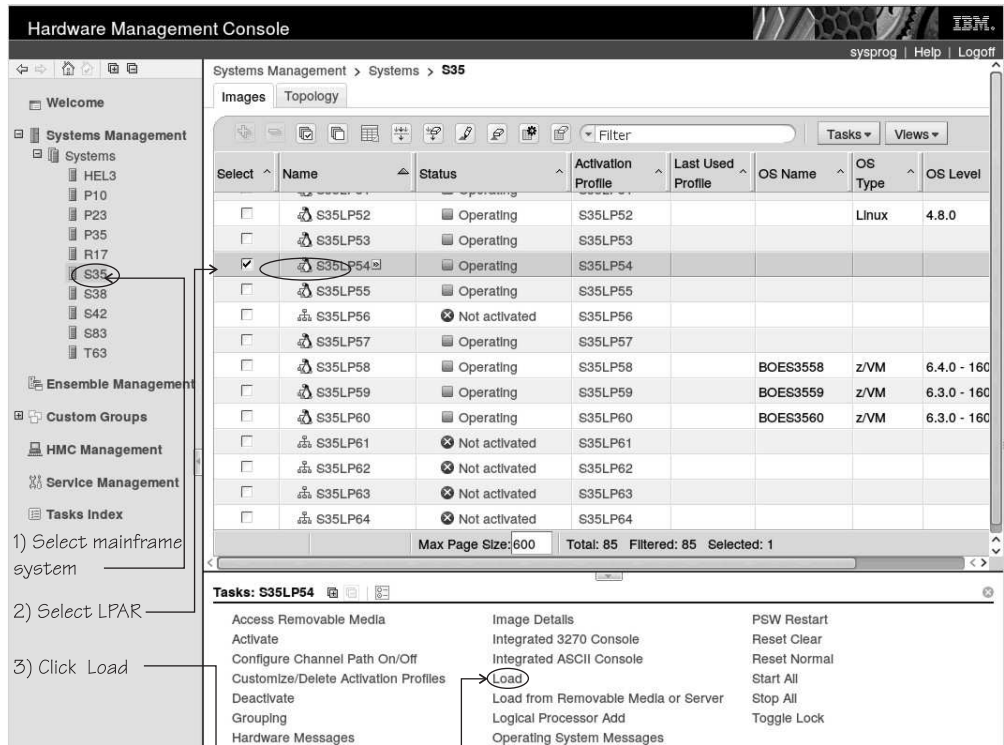


Figure 9. HMC with the **Load** task

4. Select **Load** type "SCSI dump".
5. Type the device number of the FCP adapter for the SCSI disk into the **Load address** field.
6. Type the World Wide Port name of the SCSI disk into the **World wide port name** field.
7. Type the Logical Unit Number of the SCSI disk into the **Logical unit number** field.
8. Type 0 in the Boot program selector field. The 'configuration number' defines the IPL or dump configuration which is to be IPLed. The numbering starts with 1 and is related to the menu of IPL/dump entries in the **zipl** configuration file for the SCSI disk. Configuration number 0 specifies the default configuration.
9. Accept the defaults for the remaining fields.

In rare cases, you might want to overwrite or complement the existing SCSI dump tools parameters that have been configured with **zipl**. For example, you might want to change the dump mode setting. In the **Operating system specific load parameters** field, you can specify SCSI dump tools parameters to be concatenated to the existing parameters.

In contrast to SCSI IPL configurations, where you can use a leading equal sign to replace all kernel parameters, you cannot use a leading equal sign to replace all SCSI dump tool parameters. Specifying the parameters with a leading equal sign causes the dump to fail. Figure 10 on page 35 shows a Load panel with all entries and selections required to start the SCSI dump process.

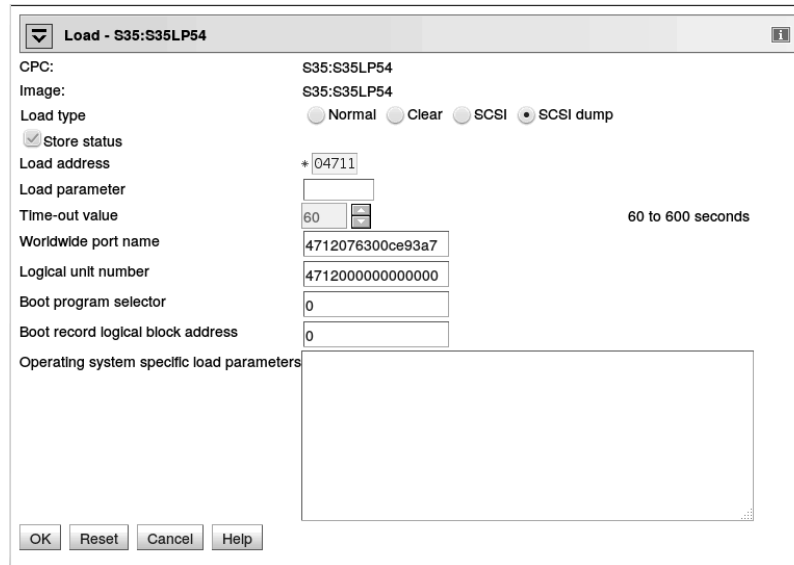


Figure 10. Load panel with enabled SCSI feature for dumping to SCSI disk

10. Click **OK** to start the dump process.
11. Wait until the dump process completes. Click the **Operating System Messages** icon for progress and error information.

## Results

When the dump has completed successfully, you can IPL Linux again.

## Automatic dump example

On both z/VM and LPAR, you can use the dumpconf service to set up automatic dumping. In this example, a dump is automatically triggered when a kernel panic occurs.

### About this task

To set up dumping using dumpconf, edit the configuration file `/etc/sysconfig/dumpconf`.

### Example

Example configuration for an FCP dump device (SCSI disk), where the disk has device bus ID 0.0.4711, the WWPN is 0x4712076300ce93a7, and the LUN is 0x4712000000000000:

```
ON_PANIC=dump
DUMP_TYPE=fcp
DEVICE=0.0.4711
WWPN=0x4712076300ce93a7
LUN=0x4712000000000000
BOOTPROG=0
BR_LBA=0
```

For details on how to set up dumpconf, see “dumpconf - Configure panic or PSW restart action” on page 56.

---

## Printing the SCSI dump header

To print the dump file header, use **zgetdump** with the **-i** option.

### Procedure

Specify the **zgetdump** command with the **-i** option:

```
# zgetdump -i /dumps/mydumps/dump.0
General dump info:
Dump format.....: lkcd
Version.....: 8
UTS node name.....: mylnxsys
UTS kernel release.: 3.10.0-685.el7.s390x
UTS kernel version.: #1 SMP Tue Jun 20 00:14:37 EDT 2017
System arch.....: s390x (64 bit)
CPU count (online).: 4
CPU count (real)...:4
Dump memory range..: 2048 MB

Memory map:
0000000000000000 - 000000007fffffff (2048 MB)
```

---

## Chapter 7. Creating dumps on z/VM with VMDUMP

Use VMDUMP to create dumps on z/VM systems, using the z/VM reader as the dump medium.

### Before you begin

Do not use VMDUMP to dump large z/VM guests; the dump process is very slow. Dumping 1 GB of storage can take up to 15 minutes depending on the used storage server and z/VM version.

### About this task

This section describes how to create a dump with VMDUMP, how to transfer the dump to Linux, and how to convert the z/VM dump to a convenient format.

VMDUMP does not need to be installed separately.

---

## Initiating a dump with VMDUMP

Start the dump VMDUMP process with the CP VMDUMP command.

### Procedure

Issue the following command from the 3270 console of the z/VM guest virtual machine:

```
#CP VMDUMP
```

### Results

z/VM CP temporarily stops the z/VM guest virtual machine and creates a dump file. The dump file is stored in the reader of the z/VM guest virtual machine. After the dump is complete, the Linux on z/VM instance continues operating.

You can use the **TO** option of the VMDUMP command to direct the dump to the reader of another guest virtual machine of the same z/VM system.

### Example

To write the dump to the reader of z/VM guest virtual machine linux02 issue:

```
#CP VMDUMP TO LINUX02
```

For more information about VMDUMP refer to *z/VM CP Commands and Utilities Reference*, SC24-6175.

## z/VM guest example

You can initiate a dump for Linux instances running under z/VM by using VMDUMP.

## Procedure

To initialize a dump with **VMDUMP**, issue this command from the console of your z/VM guest virtual machine:

```
#cp vmdump
```

## Results

Dumping does not force you to perform an IPL. If the Linux instance ran as required before dumping, it continues running after the dump is completed.

---

## Copying the dump to Linux

Copy the dump from the z/VM reader using the **vmur** command.

### Procedure

1. Find the spool ID of the **VMDUMP** spool file in the output of the **vmur li** command:

```
# 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
```

In the example the required **VMDUMP** file spool ID is 463.

2. Copy the dump into your Linux file system using the **vmur receive** command. To convert the dump into a format that can be processed with the Linux dump analysis tool **crash**, convert the dump using the **--convert** option:

```
# vmur rec 463 -c myvmdump
vmdump information:
architecture: 64 bit (big)
storage.....: 256 MB
date.....: Thu Feb 5 08:39:48 2009
cpus.....: 1
256 of 256 |#####| 100%
```

## Results

The created file, named **myvmdump**, can now be used as input to **crash**.



---

## Chapter 8. Creating live-system dumps with zgetdump

If you require a kernel dump of a Linux instance, but no downtime is acceptable, you can create a kernel dump from a live system without disruption.

Because the Linux system continues running while the dump is written, and kernel data structures are changing during the dump process, the resulting dump contains inconsistencies. The faster the dump process completes, the fewer inconsistencies the resulting live-system dump will contain. Therefore, run the dump process with the highest acceptable priority.

You can change the scheduling priority with the `nice` command. For example, use `nice -n -20` to set the highest possible priority.

---

### Creating a kernel dump on a live system

You can create non-disruptive kernel dumps on a running Linux system with the `zgetdump` tool.

#### Before you begin

- The dump directory needs enough free space (memory size + 10 MB) to hold the system memory.
- Ensure that during the dump process, no memory hotplug or CPU hotplug is performed.
- If applicable, stop the `cpuplugd` service by issuing the command:

```
# service cpuplugd stop
```

- Load the crash kernel module by issuing the command:

```
# modprobe crash
```

#### Procedure

1. Optional: Use the `-i` option to print information for the currently running Linux image:

```
# zgetdump -i /dev/crash
General dump info:
  Dump format.....: devmem
  Dump method.....: live
  UTS node name.....: mylnxsys
  UTS kernel release.: 3.10.0-685.e17.s390x
  UTS kernel version.: #1 SMP Tue Jun 20 00:14:37 EDT 2017
  System arch.....: s390x (64 bit)
  Dump memory range.: 896 MB

Memory map:
0000000000000000 - 0000000037ffffff (896 MB)
```

2. Create a dump from a live system by specifying `/dev/crash` as input dump and redirecting the output to a dump file. Run the dump process with a high priority.

```
# nice -n -20 zgetdump /dev/crash > dump.s390
```

Optionally, you can also specify a target dump format with the **-f** option:

```
# zgetdump /dev/crash -f elf > dump.elf
```

3. Optional: Print information for the live-system dump. Use the **-i** option to print information for live-system dumps that are generated by **zgetdump**:

```
# zgetdump -i dump.elf
General dump info:
Dump format.....: elf
Version.....: 1
Dump method.....: live
UTS node name.....: mylnxsys
UTS kernel release.: 3.10.0-685.el7.s390x
UTS kernel version.: #1 SMP Tue Jun 20 00:14:37 EDT 2017
System arch.....: s390x (64 bit)
Dump memory range.: 896 MB

Memory map:
0000000000000000 - 0000000037ffffff (896 MB)
```

The value "live" in the **Dump method** field indicates that this is a dump from a live system.

## Example

```
# nice -n -20 zgetdump /dev/crash -f elf > dump.elf
Format Info:
Source: devmem
Target: elf
Copying dump:
00000000 / 00000896 MB
00000149 / 00000896 MB
...
00000747 / 00000896 MB
00000896 / 00000896 MB
Success: Dump has been copied
```

## What to do next

After you create a dump from a live system, you can work with crash, see “Opening a live-system dump with the crash tool.”

After the live dump has been copied to a file system, you can compress it with **makedumpfile**. Note that the dump level must not be greater than 1 because of the dump inconsistencies.

For example:

```
# makedumpfile dump.elf -c -d 1 dump.kdump
```

---

## Opening a live-system dump with the crash tool

Inconsistencies in a kernel dump from a live system can cause some crash commands to fail.

### Before you begin

You might need to install debuginfo to see information about a dump when you use the **crash** command. Use the **yum** command to install debuginfo, for example:

```
# yum install kernel-debuginfo
```

Should you need to install crash, you can use yum again, for example:

```
# yum install crash
```

## Procedure

- Use the **crash** command to find information about whether a dump is from a live system. This information is displayed in the startup messages, or when you use the **sys** command:

```
# crash dump.elf /usr/lib/debug/lib/modules/3.10.<xx>.e17.s390x/vmlinux
...
  KERNEL: /usr/lib/debug/lib/modules/3.10.<xx>.e17.s390x/vmlinux
  DUMPFILE: dump.elf [LIVE DUMP]
  CPUS: 6
...
crash> sys | grep DUMPFILE
...
  DUMPFILE: dump.elf [LIVE DUMP]
...
```

The tag [LIVE DUMP] informs you that the dump contains inconsistencies.

- Detect whether a dump is from a live system by using the **help -p** command:

```
# crash> help -p | grep flags2
flags2: 40 (LIVE_DUMP)
```

- Use the **--minimal** option if the crash tool fails to start because of inconsistent data structures in the kernel dump. With this option, crash tolerates a degree of inconsistency. However, only a subset of crash commands is then available:

```
# crash --minimal dump.elf /usr/lib/debug/lib/modules/3.10.<xx>.e17.s390x/vmlinux
...
NOTE: minimal mode commands: log, dis, rd, sym, eval, set and exit
```



---

## Chapter 9. Processing dumps

You can copy and transfer the dump file to another system, reduce the dump size, and send the reduced dump to IBM Support.

### Procedure

- To copy and transfer an LPAR dump on DASD, tape, or SCSI dump to another system, use the **zgetdump** command, see “zgetdump - Copy and convert kernel dumps” on page 51. With **kdump**, you can transmit the dump through a network. Use existing mechanisms to prevent conflicts when concurrently writing multiple dumps to a shared persistent storage space.
- To analyze the dump, you can use **crash**, see “crash - Analyze kernel dumps” on page 59.
- To receive a VMDUMP file, use the **vmur** command, see “vmur - Receive dumps from the z/VM reader” on page 60

---

## Reducing dump size

Methods exist for handling memory dumps that are especially large (greater than 10 GB in size).

### Before you begin

The preferred method for handling memory dumps of large production systems is using **kdump**. With **kdump**, you do not need to set up a dedicated dump device with a dump tool for each individual system. Instead, you set aside storage space to receive any dumps from across your installation. If you want to set up a backup dump method for a critical system with a large memory, the information here applies when you use **kdump**.

### About this task

Large memory dumps present a challenge as they:

- Take up a large amount of disk space
- Take a long time when dumping
- Use considerable network bandwidth when they are sent to the service organization.

**Note:** Sometimes you can re-create the problem on a test system with less memory, which makes the handling much easier. Take this option into account before you create a large memory dump.

### Procedure

Complete these steps to prepare and process a large memory dump.

1. Choose a dump device. If you want to dump a system with a large memory footprint, you must prepare a dump device that is large enough. You can use the following dump devices for large dumps:

#### Single-volume DASD

- 3390 model 9 (up to 45 GB)

- 3390 model A (up to 1 TB)

### Multivolume DASD

Up to 32 DASDs are possible.

- 32 x 3390 model 9 (up to 1.4 TB)
- 32 x 3390 model A (up to 32 TB)

### z/VM emulated FBA device that represents a real SCSI device

FBA disks can be defined with the CP command **SET EDEVICE**. These disks can be used as single-volume DASD dump disks. The SCSI disk size depends on your storage server setup.

### SCSI disk

The SCSI disk size depends on your storage server setup. The ext4 file system dump size limit is 16 TB and for xfs it is 8 EB. For SCSI dump partitions greater than 2 TB, you must use the GPT disk layout.

### Dump on 3592 channel-attached tape drive

Cartridges with up to 10TB capacity.

Do not use **VMDUMP** for large systems because this dump method is very slow.

2. Estimate the dump time. The dump speed depends on your environment, for example your SAN setup and your storage server. Assume about 100 MB per second dump speed on DASDs or SCSI disks and a system with 50 GB memory. The dump will take about eight minutes. Do a test dump on your system to determine the dump speed for it. Then, you have an indication of how long a dump takes in case of emergency.
3. Reduce the dump size. For transferring dumps in a short amount of time to a service organization, it is often useful to reduce the dump size or split the dump into several parts for easier and faster transmission. To reduce the dump, choose one of these methods:
  - “Compressing a dump using `makedumpfile`”
  - “Compressing a dump using `gzip` and `split`” on page 45
4. Send the dump.

## Compressing a dump using `makedumpfile`

Use the `makedumpfile` tool to compress s390 dumps and exclude memory pages that are not needed for analysis. Alternatively, you can use the `gzip` and `split` commands.

### About this task

Compressing the dump substantially reduces the size of dump files and the amount of time needed to transmit them from one location to another. For Red Hat Enterprise Linux, the `makedumpfile` tool is included in the `kexec-tools` RPM that you can install, for example, with `yum install kexec-tools`. Because `makedumpfile` expects as input dump files in ELF format, you first have to transform your s390 format dump to ELF format. This is best done by mounting the dump using the `zgetdump` command.

### Procedure

1. Mount the dump in ELF format by performing one of these steps:
  - To mount a DASD dump from the partition `/dev/dasdb1` to `/mnt`, issue:

```
# zgetdump -m -f elf /dev/dasdb1 /mnt
```

- To mount a SCSI dump from file `dump.0` to `/dumps/mydumps`, issue:

```
# zgetdump -m -f elf dump.0 /dumps/mydumps
```

2. Locate the `vmlinux` file in the `debuginfo` RPM. After mounting the dump in ELF format with `zgetdump`, the dump is available in the file named `/mnt/dump.elf`. In order to use `makedumpfile` with dump level greater than one, you also need the `vmlinux` file that contains necessary debug information. You find this file in the kernel `debuginfo` RPM. Issue the following commands (the `xx` in the example must be replaced by the appropriate kernel version that caused the dump):

```
# rpm -qlp kernel-debuginfo-3.10-xx.e17.s390x.rpm | grep vmlinux
```

3. Extract the `vmlinux` file to `./usr/lib/debug/lib/modules/3.10-xx.e17.s390x/`. Issue the following command:

```
# rpm2cpio kernel-debuginfo-3.10-xx.e17.s390x.rpm | cpio -idv *vmlinux*
./usr/lib/debug/lib/modules/3.10-xx.e17.s390x/vmlinux
1079519 blocks
```

4. Use the `-d` (dump level) option of `makedumpfile` to specify which pages to exclude from the dump. See the man page for `makedumpfile` for a description of the dump level and other options of `makedumpfile`.

This example compresses the dump file named `/mnt/dump.elf` (`-c` option) and excludes pages that are typically not needed to analyze a kernel problem. Excluded pages are: pages containing only zeroes, pages used to cache file contents (cache, cache private), pages belonging to user spaces processes, and free pages (maximum dump level 31):

```
# makedumpfile -c -d 31 -x ./usr/lib/debug/lib/modules/3.10-xx.e17.s390x/vmlinux \
/mnt/dump.elf dump.kdump
```

The newly created file, named `dump.kdump` should be much smaller than the original file, named `dump.elf`. Until your kernel problem is resolved, it is recommended to keep the original dump file. This will enable you to reduce the dump level, if it turns out that the pages that had been excluded are still needed for problem determination.

5. For initial problem analysis, you can also extract the kernel log with `makedumpfile`, and send it to your service organization:

```
# makedumpfile --dump-dmesg -x ./usr/lib/debug/lib/modules/3.10-xx.e17.s390x/vmlinux \
/mnt/dump.elf kernel.log
```

## What to do next

After you have used `makedumpfile`, you can unmount the dump:

```
# zgetdump -u /mnt
```

## Compressing a dump using `gzip` and `split`

Use the `gzip` and `split` commands to compress the dump and split it into parts. Alternatively, you can use the `makedumpfile` command.

## Procedure

1. Compress the dump and split it into parts of one GB by using the **gzip** and **split** commands.

- For a DASD dump:

```
# zgetdump /dev/dasdd1 | gzip | split -b 1G
```

- For a tape dump:

```
# mt -f /dev/ntibm0 rewind
# mt -f /dev/ntibm0 fsf
# zgetdump /dev/ntibm0 | gzip | split -b 1G
```

- For a SCSI dump:

```
# cat /dumps/mydumps/dump.0 | gzip | split -b 1G
```

This creates several compressed files in your current directory:

```
# ls
# xaa xab xac xad xae
```

2. Create md5 sums of parts:

```
# md5sum * > dump.md5
```

3. Upload the parts together with the MD5 information to the service organization.
4. The receiver (the service organization) must do the following:
  - a. Verify md5 sums:

```
# cd dumpdir
# md5sum -c dump.md5
xaa: OK
xab: OK
...
```

- b. Merge parts and uncompress the dump:

```
# cat x* | gunzip -c > dump
```

---

## Preparing for analyzing a dump

To analyze your dump with **crash**, additional files are required.

If you need to send your dump for analysis, it might be good to include these additional files with the dump file. Your distribution typically provides the additional files in RPMs.

To begin analyzing a Red Hat Enterprise Linux 7.4 dump with **crash**, this file (as a minimum) is required:

- **vmlinux** (full): Contains addresses of kernel symbols and datatype debug information

If you need to send your dump for analysis, include the RPMs shown in Table 3 on page 47.



Table 3 lists RPMs that contain important files for debugging kernel dumps with **crash**.

*Table 3. Red Hat Enterprise Linux 7.4 debug files, paths, and RPMs*

Debug files	Path	RPM
vmlinux (full)	/usr/lib/debug/lib/modules/3.10.0-xx.el7.s390x/vmlinux	kernel-debuginfo-<version>.el7.s390x.rpm
modules (text)	lib/modules/3.10.0-xx.el7.s390x/kernel/	kernel-xx.el7.s390x.rpm
modules (debug)	usr/lib/debug/lib/modules/3.10.0-xx.el7.s390x/kernel/	kernel-debuginfo-xx.el7.s390x.rpm
debug source	usr/src/debug/kernel-3.10.0-xx.el7/linux-3.10.0-xx.el7.s390x/	kernel-debuginfo-common-s390x-xx.el7.s390x.rpm
trace events	usr/lib/debug/usr/lib64/traceevent/plugins	kernel-debuginfo-common-s390x-xx.el7.s390x.rpm

Files that contain the addresses of kernel symbols are named (text).

Files that contain the corresponding datatype debug information are named (debug).

Files that contain both are named (full).

---

## Sending a dump to IBM Support

After compressing the dump, you can transfer it using FTP or an HTTPS upload server.

### Before you begin

You need to have opened a problem management record (PMR) before you send the data. You need the PMR number to correctly name the uploaded files.

### Procedure

See Enhanced Customer Data Repository at <http://www.ibm.com/de/support/ecurep/index.html> for detailed instructions.



---

## Appendix A. Obtaining a dump with limited size

The `mem` kernel parameter can make Linux use less memory than is available to it. A dump of such a Linux system does not need to include the unused memory. You can use the `zipl size` option to limit the amount of memory that is dumped.

### About this task

The `size` option is available for all `zipl` based dumps: DASD, tape, and SCSI, in command-line mode or in configuration-file mode. The `size` option is appended to the dump device specification with a comma as separator.

The value is a decimal number that can optionally be suffixed with K for kilobytes, M for megabytes, or G for gigabytes. Values specified in byte or kilobyte are rounded to the next megabyte boundary.

Be sure not to make the dump size smaller than the amount of memory actually used by the system to be dumped. Limiting the dump size to less than the amount of used memory results in an incomplete dump.

### Example

The following command prepares a DASD dump device for a dump that is limited to 100 megabyte:

```
# zipl -d /dev/dasdc1,100M
```



---

## Appendix B. Command summary

The descriptions of the commands contain only the relevant options and parameters, for a full description refer to the man pages.

- “zgetdump - Copy and convert kernel dumps”
- “dumpconf - Configure panic or PSW restart action” on page 56
- “crash - Analyze kernel dumps” on page 59
- “vmconvert - Convert z/VM VMDUMPS for Linux” on page 59
- “vmur - Receive dumps from the z/VM reader” on page 60

---

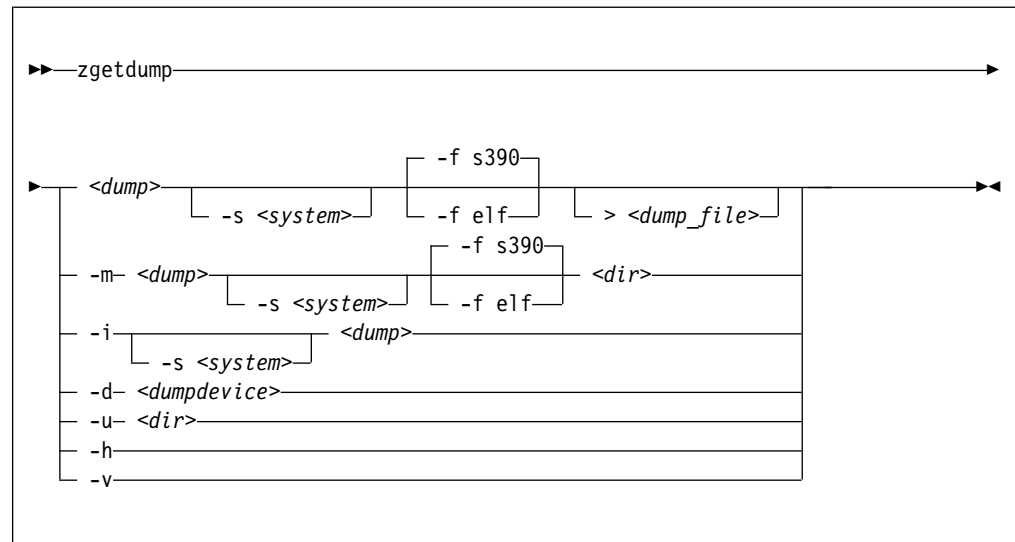
### zgetdump - Copy and convert kernel dumps

The **zgetdump** tool reads or converts a dump.

The dump can be located either on a dump device or on a file system. The dump content is written to standard output, unless you redirect it to a specific file. You can also mount the dump content, print dump information, or check whether a DASD device contains a valid dump tool.

**Before you begin:** Mounting is implemented with "fuse" (file system in user space). Therefore the fuse kernel module must be loaded before you can use the **--m** option. Load the module, for example, with **modprobe fuse**.

#### zgetdump syntax



#### Parameters

##### <dump>

is the file, DASD device or partition, or channel-attached tape device node where the dump is located:

- Regular dump file (for example /testdir/dump.0)
- Dump file from the SCSI dumper (for example /dmp/mydumps/dump.0)
- DASD partition device node (for example /dev/dasdc1)

- DASD device node for multivolume dump (for example /dev/dasdc)
- Tape device node (for example /dev/ntibm0)

**Note:** For a DASD multivolume dump it is sufficient to specify only one of the multivolume DASDs as *<dump>*.

*<dump\_file>*

Is the file to which the output is redirected. The default is standard output.

*<dumpdevice>*

Specifies the dump device for the **-d** option. The device node of the DASD device, for example /dev/dasdb.

**-s** *<system>* **or** **--select** *<system>*

for dumps that capture two systems, selects the system of interest. To check whether a dump contains two systems, use **zgetdump -i**. The **-s** option is mandatory when accessing the dump of a crashed kdump instance, but returns an error if applied to a regular dump.

A dump can contain data for a crashed production system and for a crashed kdump system. A dump like this is created if a stand-alone dump tool is used to create a dump for a kdump instance that crashed while creating a dump for a previously crashed production system. *<system>* can be:

**prod**

to select the data for the crashed production system.

**kdump**

to select the data for the kdump instance that crashed while creating a dump for the previously crashed production system.

**-m** *<dump>* *<dir>* **or** **--mount** *<dump>* *<dir>*

Mounts the *<dump>* to mount point *<dir>* and generates a virtual target dump file instead of writing the content to standard output. The virtual dump file is named *dump.FMT*, where FMT is the name of the specified dump format (see the **--fmt** option).

**-u** *<dir>* **or** **--umount** *<dir>*

Unmounts the dump that is mounted at mount point *<dir>*. You can specify the dump itself instead of the directory, for example /dev/dasdd1. This option is a wrapper for **fusermount -u**.

**-i** *<dump>* **or** **--info** *<dump>*

Displays the dump header information from the dump and performs a validity check.

**-d** *<dumpdevice>* **or** **--device** *<dumpdevice>*

Checks whether the specified ECKD or FBA device contains a valid dump tool and prints information about it.

**-f** *<format>* **or** **--fmt** *<format>*

Uses the specified target dump format *<format>* when writing or mounting the dump. The following target dump formats are supported:

**elf** Executable and Linking Format core dump (64 bit only)

**s390** S/390® dump (default)

**-h** **or** **--help**

Displays the help information for the command.

**-v** **or** **--version**

Displays the version information for the command.

## Using zgetdump to copy a dump

Assuming that the dump is on DASD partition /dev/dasdb1 and that you want to copy it to a file named dump\_file:

```
# zgetdump /dev/dasdb1 > dump_file
```

## Using zgetdump to transfer a dump with ssh

Assuming that the dump is on DASD partition /dev/dasdd1 and that you want to transfer it to a file on another system with ssh:

```
# zgetdump /dev/dasdd1 | ssh user@host "cat > dump_file_on_target_host"
```

## Using zgetdump to transfer a dump with FTP

Follow these steps to transfer a dump with FTP:

1. Establish an FTP session with the target host and log in.
2. To transfer a file in binary mode, enter the FTP **binary** command:

```
ftp> binary
```

3. To send the dump file to the host issue a command of the following form:

```
ftp> put |"zgetdump /dev/dasdb1" <dump_file_on_target_host>
```

## Using zgetdump to copy a multi-volume dump

Assuming that the dump is on DASD devices /dev/dasdc and /dev/dasdd spread along partitions /dev/dasdc1 and /dev/dasdd1, and that you want to copy it to a file named multi\_volume\_dump\_file:

```
# zgetdump /dev/dasdc > multi_volume_dump_file
```

For an example of the output from this command, see Chapter 4, “Using DASD devices for multi-volume dump,” on page 19.

## Using zgetdump to copy a tape dump

Assuming that the tape device is /dev/ntimb0:

```
# zgetdump /dev/ntimb0 > dump_file
Format Info:
Source: s390tape
Target: s390

Copying dump:
00000000 / 00001024 MB
00000171 / 00001024 MB
00000341 / 00001024 MB
00000512 / 00001024 MB
00000683 / 00001024 MB
00000853 / 00001024 MB
00001024 / 00001024 MB

Success: Dump has been copied
```

## Using zgetdump to create a dump from a live system

To store an ELF-format dump from a live system in a file called `dump.elf`, issue:

```
# nice -n -20 zgetdump /dev/crash -f elf > dump.elf
```

For an example of the output from this command, see “Creating a kernel dump on a live system” on page 39.

## Checking whether a tape dump is valid, and printing the dump header

Assuming that the tape device is `/dev/ntibm0`:

```
# zgetdump -i /dev/ntibm0
Checking tape, this can take a while...
General dump info:
Dump format.....: s390tape
Version.....: 5
Dump created.....: Mon, 14 Jan 2013 17:26:46 +0200
Dump ended.....: Mon, 14 Jan 2013 17:27:58 +0200
Dump CPU ID.....: ff00012320948000
UTS kernel release.: 3.10.0-123.el7.s390x
UTS kernel version.: #1 SMP Mon Nov 19 16:52:53 EST 2012
Build arch.....: s390x (64 bit)
System arch.....: s390x (64 bit)
CPU count (online): 2
CPU count (real)...: 2
Dump memory range..: 1024 MB
Real memory range..: 1024 MB

Memory map:
0000000000000000 - 000000003fffffff (1024 MB)
```

## Checking whether a DASD dump is valid and printing the dump header

Assuming that the dump is on a partition, `part1`, of a DASD device `/dev/dasdb1`:

```
# zgetdump -i /dev/dasdb1
General dump info:
Dump format.....: s390
Version.....: 5
Dump created.....: Mon, 10 May 2010 17:32:36 +0200
Dump ended.....: Mon, 10 May 2010 17:32:48 +0200
Dump CPU ID.....: ff00012320948000
Build arch.....: s390x (64 bit)
System arch.....: s390x (64 bit)
CPU count (online): 2
CPU count (real)...: 2
Dump memory range..: 1024 MB
Real memory range..: 1024 MB

Memory map:
0000000000000000 - 000000003fffffff (1024 MB)
```



## Checking whether a SCSI dump is valid and printing the dump header

Assuming that the dump is stored in `/dumps/mydumps/dump.0`:

```
# zgetdump -i /dumps/mydumps/dump.0
General dump info:
Dump format.....: lkcd
Version.....: 8
UTS node name.....: mylnxsys
UTS kernel release.: 3.10.0-685.e17.s390x
UTS kernel version.: #1 SMP Tue Jun 20 00:14:37 EDT 2017
System arch.....: s390x (64 bit)
CPU count (online): 4
CPU count (real)...: 4
Dump memory range.: 2048 MB

Memory map:
0000000000000000 - 000000007fffffff (2048 MB)
```

## Checking whether a device contains a valid dump record

Checking DASD device `/dev/dasda`, which is a valid dump device:

```
# zgetdump -d /dev/dasdb
Dump device info:
Dump tool.....: Single-volume DASD dump tool
Version.....: 2
Architecture.....: s390x (64 bit)
DASD type.....: ECKD
Dump size limit...: none
```

Checking DASD device `/dev/dasdc`, which is not a valid dump device:

```
# zgetdump -d /dev/dasdc
zgetdump: No dump tool found on "/dev/dasdc"
```

## Using the mount option

Mounting is useful for multivolume DASD dumps. After a multivolume dump has been mounted, it is shown as a single dump file that can be accessed directly with dump processing tools such as **crash**.

The following example mounts a multivolume DASD dump as an ELF dump, processes it with **crash**, and unmounts it with **zgetdump**:

```
# zgetdump -m -f elf /dev/dasdx /dumps
# crash vmlinux /dumps/dump.elf
# zgetdump -u /dumps
```

Mounting can also be useful when you want to process the dump with a tool that cannot read the original dump format. To do this, mount the dump and specify the required target dump format with the `--fmt` option.

## Selecting data from a dump that includes a crashed kdump

The following example mounts dump data for a crashed production system from a DASD backup dump for a failed kdump (see “Failure recovery and backup tools” on page 9 for details).

```
# zgetdump -s prod -m /dev/dasdb1 /mnt
```

## Checking whether a dump has captured two systems

A dump can contain data from two systems. To check for this use **zgetdump -i**, for example, assuming that the previous dump example contains both a dump from the production system and a kdump kernel dump:

```
# zgetdump -i /dumps/mydumps/dump.0
zgetdump: The dump contains "kdump" and "production system"
Access "production system" with "-s prod"
Access "kdump" with "-s kdump"
Send both dumps to your service organization
```

---

## dumpconf - Configure panic or PSW restart action

The **dumpconf** service configures the action to be taken if a kernel panic or PSW restart occurs.

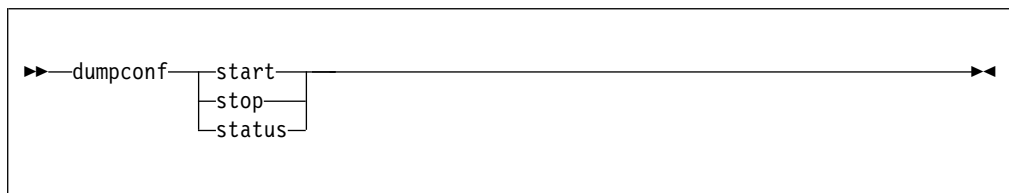
The service is installed as a script under `/etc/init.d/dumpconf` and reads the configuration file `/etc/sysconfig/dumpconf`.

**Note:** kdump does not depend on **dumpconf** and can neither be enabled nor disabled with **dumpconf**. If kdump has been set up for your production system, dump tools as configured with **dumpconf** are used only if the integrity check for kdump fails. With kdump set up, you can use **dumpconf** to enable or disable backup dump tools. See also “Failure recovery and backup tools” on page 9.

To enable the **dumpconf** service, issue:

```
# chkconfig --add dumpconf
```

### dumpconf service syntax



### Parameters

#### start

Enable configuration defined in `/etc/sysconfig/dumpconf`.

#### stop

Disable the **dumpconf** service.

#### status

Show current configuration status of the **dumpconf** service.

#### -h or --help

Display short usage text on console. To view the man page, enter **man dumpconf**.

**-v or --version**

Display version number on console, and exit.

## Keywords for the configuration file

### ON\_PANIC

Shutdown action to be taken if a kernel panic or PSW restart occurs. Possible values are:

**dump** Dump Linux and stop system.

**reipl** Reboot Linux.

**dump\_reipl**  
Dump Linux and reboot system.

**vmcmd**  
Execute specified CP commands and stop system.

**stop** Stop Linux (default).

### DELAY\_MINUTES

The number of minutes that the activation of the **dumpconf** service is to be delayed. The default is zero.

Using **reipl** or **dump\_reipl** actions with **ON\_PANIC** can lead to the system looping with alternating IPLs and crashes. Use **DELAY\_MINUTES** to prevent such a loop. **DELAY\_MINUTES** delays activating the specified panic action for a newly started system. When the specified time has elapsed, the **dumpconf** service activates the specified panic action. This action is taken should the system subsequently crash. If the system crashes before the time has elapsed, the previously defined action is taken. If no previous action has been defined, the default action (STOP) is performed.

### VMCMD\_<X>

Specifies a CP command, <X> is a number from one to five. You can specify up to five CP commands that are executed in case of a kernel panic or PSW restart. Note that z/VM commands, device addresses, and names of z/VM guest virtual machines must be uppercase.

### DUMP\_TYPE

Type of dump device. Possible values are **ccw** and **fcp**.

### DEVICE

Device bus-ID of the dump device.

### WWPN

WWPN for SCSI dump device.

**LUN** LUN for SCSI dump device.

### BOOTPROG

Boot program selector

### BR\_LBA

Boot record logical block address.

## Example configuration files for the dumpconf service

- Example configuration for a CCW dump device (DASD) using **reipl** after dump and **DELAY\_MINUTES**:

```
ON_PANIC=dump_reipl
DUMP_TYPE=ccw
DEVICE=0.0.4714
DELAY_MINUTES=5
```

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

```
ON_PANIC=dump
DUMP_TYPE=fcp
DEVICE=0.0.4711
WWPN=0x5005076303004712
LUN=0x4713000000000000
BOOTPROG=0
BR_LBA=0
```

- Example configuration for re-IPL if a kernel panic or PSW restart occurs:

```
ON_PANIC=reipl
```

- Example of sending a message to the z/VM guest virtual machine "MASTER", executing a **CP VMDUMP** command, and rebooting from device 4711 if a kernel panic or PSW restart occurs:

```
ON_PANIC=vmcmd
VMCMD_1="MSG MASTER Starting VMDUMP"
VMCMD_2="VMDUMP"
VMCMD_3="IPL 4711"
```

Note that z/VM commands, device addresses, and names of z/VM guest virtual machines must be uppercase.

## Examples for using the dumpconf service

Use the **dumpconf** service to enable and disable the configuration.

- To enable the configuration:

```
# service dumpconf start
ccw dump device configured. "dump" on panic configured.
```

- To display the status:

```
# service dumpconf status
type....: ccw
device..: 0.0.4714
on_panic: dump
```

- To disable dump on panic:

```
# service dumpconf stop
Dump on panic is disabled now
```

- To display the status again and check that the status is now stopped.

```
# service dumpconf status
on_panic: stop
```

---

## crash - Analyze kernel dumps

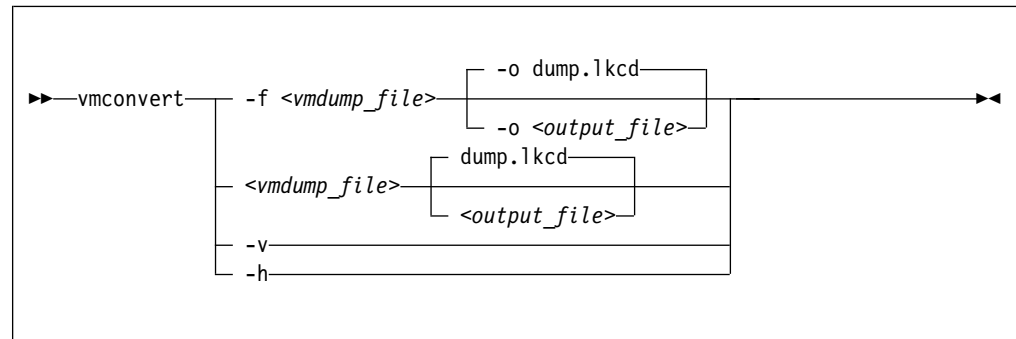
The **crash** tool is a GPL-licensed tool maintained by Red Hat. For more details see the tool online help.

---

## vmconvert - Convert z/VM VMDUMPS for Linux

The **vmconvert** tool converts a dump that was created with **VMDUMP** into a file that can be analyzed with **crash**.

### vmconvert syntax



### Parameters

**<vmdump\_file>** or **-f <vmdump\_file>** or **--file <vmdump\_file>**

Specifies the **VMDUMP** created dump file to be converted.

**<output\_file>** or **-o <output\_file>** or **--output <output\_file>**

Specifies the name of the dump file to be created. The default is `dump.lkcd`.

**-v** or **--version**

Displays the tool version.

**-h** or **--help**

Displays the help information for the command.

### Example

To convert a **VMDUMP**-created dump file `vmdump1` into a dump file `dump1.lkcd` that can be processed with **crash**, issue:

```
# vmconvert -f vmdump1 -o dump1.lkcd
```

You can also use positional parameters:

```
# vmconvert vm.dump lkcd.dump
vmdump information:
architecture: 32 bit
date.....: Fri Feb 18 11:06:45 2005
storage.....: 16 MB
cpus.....: 6
16 of 16 |#####| 100%
'lkcd.dump' has been written successfully.
```

---

## vmur - Receive dumps from the z/VM reader

The **vmur** command can receive a **VMDUMP** file from the z/VM reader and convert it into a file that can be analyzed with **crash**.

Issue a command of the following form:

```
# vmur receive -c <spool ID> <dump file name>
```

### Parameters

**<spool ID>**

Specifies the **VMDUMP** file spool ID.

**<dump file name>**

Specifies the name of the output file to receive the reader spool file's data.

For more details, see the **vmur** man page and *Device Drivers, Features, and Commands on Red Hat Enterprise Linux 7.4*, SC34-2710

### Example

#### Before you begin:

The **vmur** commands access the reader device, which has to be online. To set it online, it needs to be freed from **cio\_ignore**:

```
# cio_ignore -r c
# chccwdev -e c
Setting device 0.0.000c online
Done
```

To receive and convert a **VMDUMP** spool file with spool ID 463 to a file named **dump.1kcd** on the Linux file system in the current working directory:

```
# vmur rec -c 463 dump.1kcd
```

---

## Accessibility

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

### Documentation accessibility

The Linux on z Systems publications are in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when you use the PDF file and want to request a Web-based format for this publication, use the Readers' Comments form in the back of this publication, send an email to [eservdoc@de.ibm.com](mailto:eservdoc@de.ibm.com), or write to:

IBM Deutschland Research & Development GmbH  
Information Development  
Department 3282  
Schoenaicher Strasse 220  
71032 Boeblingen  
Germany

In the request, be sure to include the publication number and title.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

### IBM and accessibility

See the IBM Human Ability and Accessibility Center for more information about the commitment that IBM has to accessibility at

[www.ibm.com/able](http://www.ibm.com/able)





---

## Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

Adobe is either a registered trademark or trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

---

# Index

## Special characters

/etc/init.d/dumpconf  
dumpconf configuration file 56  
/etc/kdump.conf  
configuration file for kdump 9

## A

accessibility 61  
analyzing dump  
preparing for 46  
authority  
root vii  
automatic dump  
dump-on-panic 5  
automatic dump-on-panic 5

## C

commands  
crash 59  
dumpconf 56  
summary 51  
vmconvert 59  
vmur 60  
zgetdump 51  
compressing dump  
using gzip and split 46  
using makedumpfile 44  
configuration file  
dumpconf 56  
kdump 9  
crash 59  
live dump, opening 40  
preparing for analyzing a dump  
with 46  
crashkernel=  
kernel parameter 7  
creating dumps  
tools overview 1

## D

DASD devices  
using as dump device 13  
using for multi-volume dump 19  
DASD dump  
initiating 14  
DASD dump tool 20  
installing 13  
DASD multi-volume dump  
starting 21  
developerWorks 3  
dump  
compressing using gzip and split 46  
compressing using makedumpfile 44  
copy from DASD with zgetdump  
command 17

dump (*continued*)  
copy multi-volume dump from DASD  
with zgetdump command 22  
copy tape from 27  
copy with zgetdump command 27  
limited size 49  
live-system 3  
multi-volume 19  
preparing for analyzing 46  
sharing space for 2  
starting a multi-volume DASD 21  
tape, checking if valid 28  
tape, initializing 23  
dump devices  
DASD 13  
definition 2  
SCSI 29  
sharing between different Linux  
versions 6  
tape 23  
dump methods  
comparison 4  
dump tool  
crash 59  
DASD, installing 13  
tape, installing 23  
dump tools  
dumpconf service 56  
multi-volume, DASD 20  
stand-alone 2  
stand-alone tape 23  
summary 51  
vmconvert 59  
VMDUMP 3, 37  
vmur 60  
zgetdump 51  
dump tools overview 1  
dumpconf service 56

## E

ECuRep 47

## F

firstboot utility  
kdump setup 9

## G

gzip command 46

## H

handling large dumps 43

## I

initiate DASD dumps  
HMC or SE 15  
initiate dumps 10  
HMC or SE 11, 16, 35  
using a DASD device 14  
using VMDUMP 38  
initiate SCSI dumps  
HMC or SE 33  
initiate tape dumps  
HMC or SE 25  
initiating a dump 37  
initiating a dump to tape 24  
initiating dump  
using SCSI 32

## K

kdump  
advantages and disadvantages 7  
comparison with other dump  
methods 4  
initiate 10  
initiating 10  
introduction 2  
setup 9  
testing automatic dump-on-panic 5  
kdump kernel 7  
kernel dump  
creating from live system 39  
Kernel Dump Configuration utility  
kdump setup 9

## L

large dump  
handling 43  
multi-volume 19  
limit amount of memory dumped 49  
Linux versions  
sharing dump devices between 6  
live dump  
open with crash 40  
live system  
creating dump 39  
live-system dump 3

## M

makedumpfile 44  
memory  
reserved for kdump kernel 7  
memory dump  
compressing 44  
compressing, using gzip and split 46  
multi-volume dump  
starting 21  
multi-volume dumps 19  
DASD tool 20

## N

- non-disruptive dump
  - using zgetdump 39

## O

- opening live-system dump
  - using crash 40

## P

- parameters
  - SCSI dump tool 29
- PMR 47
- profs
  - required authority vii

## R

- root authority vii

## S

- s390utils 2
- SCSI dump
  - initiating 31
  - printing the dump header 36
  - single partition 30
- SCSI dump device 29
- SCSI dump tool
  - installing 29
  - parameters 29
- SCSI Linux System Dumper 29
- SD
  - See SCSI Linux System Dumper
- sending a dump to IBM 47
- single partition
  - used for SCSI dump 30
- split command 46
- summary
  - commands for dumps 51
- system restart
  - z/VM CP command 10

## T

- tape 24
  - copy dump from 27
  - use for dumping, preparing 27
  - using as dump device 23
- tape dump
  - checking if valid 28
  - initializing 23
- tape dump tool
  - installing 23
- testing 5
- tools for creating dumps 1
- transfer time
  - reducing with kdump 2

## U

- using kdump 10

## V

- vmconvert 59
- VMDUMP 37
  - comparison with other dump
    - methods 4
  - copying dump 38
  - initiate dump process 37
  - introduction 3
- vmur 60
  - using to copy VMDUMP dump 38

## Z

- z/VM CP command
  - system restart 10
- zgetdump 51
  - comparison with other dump
    - methods 4
  - create a dump from a live system 39
- zgetdump tool 17, 22, 27
- zipl
  - size option 49

---

## Readers' Comments — We'd Like to Hear from You

Linux on z Systems and LinuxONE  
Using the Dump Tools  
on Red Hat Enterprise Linux 7.4

Publication No. SC34-2711-02

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via email to: S390ID@de.ibm.com

If you would like a response from IBM, please fill in the following information:

\_\_\_\_\_

Name

\_\_\_\_\_

Address

\_\_\_\_\_

Company or Organization

\_\_\_\_\_

Phone No.

\_\_\_\_\_

Email address



Fold and Tape

**Please do not staple**

Fold and Tape

PLACE  
POSTAGE  
STAMP  
HERE

IBM Deutschland Research & Development GmbH  
Information Development  
Department 3282  
Schoenaicher Strasse 220  
71032 Boeblingen  
Germany

Fold and Tape

**Please do not staple**

Fold and Tape





SC34-2711-02

