Using the Dump Tools on SUSE Linux Enterprise Server 11 SP1
Using the Dump Tools on SUSE Linux Enterprise Server 11 SP1
Note

Before using this information and the product it supports, read the information in "Notices" on page 39.

This edition applies to the Novell SUSE Linux Enterprise Server 11 SP1 on IBM System z, and to all subsequent releases and modifications until otherwise indicated in new editions.

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

**About this document** ................................................................. v
Other Linux on IBM System z publications for SLES 11 SP1 ........................ v

**Chapter 1. Introduction** .......................................................... 1
Stand-alone tools .......................................................................... 2
VMDUMP ......................................................................................... 2

**Chapter 2. Using a DASD dump device** ........................................... 3
Installing the DASD dump tool ...................................................... 3
Installing the DASD dump tool using Novell Yast ............................. 4
Initiating a DASD dump .................................................................... 4
Copying the dump from DASD with zgetdump .................................. 5

**Chapter 3. Using DASD devices for multi-volume dump** .................... 7
Installing the multi-volume DASD dump tool .................................. 8
Initiating a multi-volume DASD dump ............................................. 9
Copying a multi-volume dump to a file .......................................... 10

**Chapter 4. Using a tape dump device** ............................................ 11
Installing the tape dump tool ....................................................... 11
Initiating a tape dump ...................................................................... 11
Tape display messages .................................................................... 12
Copying the dump from tape .......................................................... 12
Preparing the dump tape ................................................................ 12
Using the zgetdump tool ................................................................ 13

**Chapter 5. Using a SCSI dump device** .......................................... 15
Installing the SCSI disk dump tool ................................................. 15
SCSI dump tool parameters ........................................................ 15
Example 1: Combined dump and target partition ............................ 16
Example 2: Menu configuration with separate dump and target partitions 16
Initiating a SCSI dump ................................................................... 17
Printing the dump header .............................................................. 18

**Chapter 6. Using VMDUMP** ......................................................... 19
Initiating a dump with VMDUMP ..................................................... 19
Copying the dump to Linux ............................................................ 19
Using the vmur command .............................................................. 19
Using the DUMPLOAD command .................................................. 20

**Appendix A. Examples for initiating dumps** .................................... 21
z/VM ....................................................................................... 21
Using DASD ............................................................................... 21
Using tape .................................................................................. 21
Using SCSI .................................................................................. 21
Using VMDUMP ......................................................................... 22
HMC or SE .................................................................................. 22

**Appendix B. Obtaining a dump with limited size** ............................... 27

**Appendix C. Command summary** ................................................ 29
The zgetdump tool ....................................................................... 29
Examples ...................................................................................... 30
About this document

This book describes tools for obtaining dumps of Linux® for IBM® System z® instances running SUSE Linux Enterprise Server 11 SP1. This book describes how to use DASD, tape, and SCSI dump devices, as well as how to use VMDUMP.

Unless stated otherwise, all z/VM® related information in this book is based on the assumption that z/VM 5.3 or later is used.

In this document, System z is taken to include System z9®, System z10®, and later IBM mainframe systems.

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


Other Linux on IBM System z publications for SLES 11 SP1

Several Linux on IBM System z publications for SLES 11 SP1 are available on developerWorks.

You can find the latest versions of these publications at


- Device Drivers, Features, and Commands on SUSE Linux Enterprise Server 11 SP1, SC34-2595
- Kernel Messages on SUSE Linux Enterprise Server 11 SP1, SC34-2600

For each of the following documents, the same Web page points to the version that most closely reflects SUSE Linux Enterprise Server 11 SP1:

- How to use FC-attached SCSI devices with Linux on System z
- How to Improve Performance with PAV
- How to use Execute-in-Place Technology with Linux on z/VM
- How to Set up a Terminal Server Environment on z/VM
- libica Programmer’s Reference
Chapter 1. Introduction

Different tools can be used for obtaining dumps of Linux on IBM System z instances running SUSE SLES 11 SP1. This chapter gives an overview of those tools.

You can use the dump analysis tool **crash** (from version 3.10-13.2) 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>
<thead>
<tr>
<th>Tool</th>
<th>Stand-alone tools</th>
<th>VMDUMP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DASD</td>
<td>Multi-volume</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DASD</td>
</tr>
<tr>
<td>Environment</td>
<td>VM and LPAR</td>
<td>VM and LPAR</td>
</tr>
<tr>
<td>z/VM NSS and DCSS</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Size</td>
<td>Small, depending on disk size</td>
<td>Large, depending on SCSI disk and what other data it contains</td>
</tr>
<tr>
<td>Speed</td>
<td>Fast</td>
<td>Fast</td>
</tr>
<tr>
<td>Medium</td>
<td>ECKD™ or FBA DASD</td>
<td>ECKD DASD</td>
</tr>
<tr>
<td>Compression possible</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Disruptive</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note:
1. As of z/VM 5.4.
2. ECKD model 27, for example, provides 27 GB.
3. SCSI disks can be emulated as FBA disks. This dump method can, therefore, be used for SCSI-only VM installations.
4. IBM TotalStorage® Enterprise Tape System 3590 and IBM 3490 Magnetic Tape Subsystem offer hardware compression.
5. 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.
Stand-alone tools

Four stand-alone dump tools are shipped in the s390-tools package as part of the zipl package:

- DASD dump tool for dumps on a single DASD device
- Multi-volume 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. The dump device is the device you want to use for dumping the memory.

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 "The dumpconf tool" on page 32.

All examples for installing stand-alone tools by using a zipl configuration file assume that /etc/zipl.conf is used as the configuration file and that /etc/zipl.conf is the default configuration file.

For more information on zipl, refer to the zipl man page and to the zipl description in Device Drivers, Features, and Commands on SUSE Linux Enterprise Server 11 SP1, SC34-2595. You can find the latest version of this document on developerWorks at: www.ibm.com/developerworks/linux/linux390/development_documentation.html

VMDUMP

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 VM guests that use z/VM named saved systems (NSS) and discontiguous saved segments (DCSS).

Do not use VMDUMP to dump large 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.
Chapter 2. Using a DASD dump device

This chapter provides information on how to install the stand-alone DASD dump tool, how to use Novell Yast to perform the installation, how to perform the dump process, and how to copy the dump to a file in a Linux file system.

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

- ECKD DASDs
  - 3380
  - 3390
- FBA DASDs

Installing the DASD dump tool

Requirement: 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, you should use the multi-volume dump tool, see Chapter 3, “Using DASD devices for multi-volume dump,” on page 7.

This section describes how to install the DASD dump tool on an unused DASD partition. Dumps are written to this partition.

The examples in this section assume that /dev/dasdc is the dump device and that we want to dump to the first partition /dev/dasdc1.

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

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

1. Format your DASD with dasdfmt (ECKD only). A block size of 4 KB is recommended:
   
   Example:
   
   ```
   # dasdfmt -f /dev/dasdc -b 4096
   ```

2. Create a partition with fdasd (ECKD only). The partition must be sufficiently large (the memory size + 10 MB):
   
   Example:
   
   ```
   # fdasd /dev/dasdc
   ```

3. Install the dump tool using the zipl command. You can specify the dump device on the command line or use a configuration file.

   Command line example:
   
   ```
   # zipl -d /dev/dasdc1
   ```

   Configuration file example:
   
   a. Edit /etc/zipl.conf to add the following lines:

   ```
   [dump_dasd]
dumpto=/dev/dasdc1
   ```
b. Issue:

    # zipl dump_dasd

**Note:** When using an ECKD-type DASD formatted with the traditional Linux disk layout ldl, the dump tool must be reinstalled using zipl after each dump.

---

**Installing the DASD dump tool using Novell Yast**

It is possible to prepare ECKD single-volume DASD dump devices with the Novell Yast tool. Complete the following steps:

1. Start Yast:

    # yast

2. Ensure that the dump DASD is activated (online):
   a. Enter menu: **Hardware → DASD**.
   b. Activate the desired DASD, if it is not already activated.

3. Prepare the DASD for dump use:
   a. Enter menu: **Hardware → Dump Devices**.
   b. Select the dump DASD.
   c. If the DASD is already formatted with a file system, and you are sure that it can be overwritten, enable the checkbox:

       ![Install Dump Record Even If Disk Already Formatted]

   Otherwise, you will receive an error on the next step.
   d. Click **Next**.

4. Confirm that you want to prepare the DASD.

5. Look for error messages. If no error message is shown, your DASD device has been prepared for dump use.

---

**Initiating a DASD dump**

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 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
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

4. You can IPL Linux again.

See Appendix A, “Examples for initiating dumps,” on page 21 for more details.

---

## Copying the dump from DASD with zgetdump

This section describes how to copy a DASD dump to a file system using the zgetdump tool.

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.

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
```

You can also use zgetdump to display information about the dump. See “Checking whether a DASD dump is valid and printing the dump header” on page 32 for an example.

For general information about zgetdump, see “The zgetdump tool” on page 29 or the man page.
Chapter 3. Using DASD devices for multi-volume dump

This chapter describes how to prepare a set of ECKD DASD devices for a multi-volume dump, how to install the stand-alone dump tool on each DASD device involved, how to perform the dump process, and how to copy the dump to a file in a Linux file system.

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

- Be in subchannel set 0.
- Be formatted with the compatible disk layout (cdl, the default option when using the `dasdfmt` command.)

You can use any block size, even mixed block sizes. However, to speed up the dump process and to reduce wasted disk space, use block size 4096.

For example, Figure 1 shows three DASD volumes, dasdb, dasdc, and dasdd, with four partitions selected to contain the dump. To earmark the partition for dump, a dump signature is written to each partition.

![Figure 1](image)

**Figure 1. Three DASD volumes with four partitions for a multi-volume dump**

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

```
/dev/dasdb2
/dev/dasdb1
/dev/dasdb3
/dev/dasdc1
/dev/dasdc2
/dev/dasdd1
/dev/dasdd2
/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 will take you through the entire process of creating a multi-volume 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.

Assume that the corresponding bus IDs (as displayed by lsdasd) are 0.0.4711 and 0.0.4712, so the respective device numbers are 4711 and 4712.

1. Format both dump volumes with dasdfmt. Specify cdl (compatible disk layout), which is the default. Preferably, use a block size of 4 KB:

```
# 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 called sample_dump_conf containing the device nodes of the two partitions, separated by one or more line feed characters (0x0a). The file's contents looks as follows:

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

4. Prepare the volumes using the zipl command. You can specify the dump list on the command line or use the zipl configuration file.

**Command line example:**

```
# zipl -M sample_dump_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)?
```

**zipl configuration file example:**

a. Copy sample_dump_conf to /boot/mvdump.conf and edit /etc/zipl.conf to add the following lines:

```
[multi_volume_dump]
mvdump=/boot/mvdump.conf
```

b. Issue:

```
# zipl multi_volume_dump
```

Now the two volumes /dev/dasdc and /dev/dasdd with device numbers 4711 and 4712 are prepared for a multi-volume dump. Use the -device option of zgetdump to display information on the involved volumes:
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 --force option. The force 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 B, “Obtaining a dump with limited size,” on page 27. For more details on zipl, refer to the description of the zipl command in the Device Drivers, Features, and Commands on SUSE Linux Enterprise Server 11 SP1, SC34-2595.

**Initiating a multi-volume DASD dump**

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

1. Stop all CPUs.
2. Store status on the IPL CPU.
3. IPL the dump tool using one of the prepared volumes, either 4711 or 4712.

**Note:** Do not clear storage!

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: 4711
00000128 / 00001024 MB
00000256 / 00001024 MB
00000384 / 00001024 MB
00000512 / 00001024 MB
Dumping to: 4712
00000640 / 00001024 MB
00000768 / 00001024 MB
00000896 / 00001024 MB
000001024 / 00001024 MB
Dump successful
```
4. You can IPL Linux again.

## Copying a multi-volume dump to a file

At this point the two volumes /dev/dasdc and /dev/dasdd (with device numbers 4711 and 4712) contain the dump. Dump data is spread along partitions /dev/dasdc1 and /dev/dasdd1.

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

```
# zgetdump /dev/dasdc > multi_volume_dump_file
Dump device: /dev/dasdc

>>> Dump header information <<<
Dump created on: Wed Apr 16 09:06:01 2008

Magic number: 0xa8190173618f23fd
Version number: 4
Header size: 4096
Page size: 4096
Dumped memory: 1073741824
Dumped pages: 262144
Real memory: 1073741824
cpu id: 0xff00012320948000
System Arch: s390x (ESAME)
Build Arch: s390x (ESAME)

>>> End of Dump header <<<

Multi-volume dump: Disk 1 (of 2)
Reading dump contents from 0.0.4711................

Multi-volume dump: Disk 2 (of 2)
Reading dump contents from 0.0.4712............... 

Dump ended on: Wed Apr 16 09:07:03 2008

Dump End Marker found: this dump is valid.
```

If you want to only check the validity of the multi-volume 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 32 for an example.
Chapter 4. Using a tape dump device

This chapter provides information on how to install the stand-alone tape dump tool, how to perform the dump process, and how to copy the dump to a file in a Linux file system.

The following tape devices are supported:
- 3480
- 3490
- 3590
- 3592

Installing the tape dump tool

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

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

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 using the zipl command. You can specify the dump device on the command line or use a configuration file.

**Command line example:**
```
# zipl -d /dev/ntibm0
```

**Configuration file example:**
1. Edit `/etc/zipl.conf` to add the following lines:
   ```
   [dump_tape]
dumpdevice=/dev/ntibm0
   ```
2. Issue zipl:
   ```
   # zipl dump_tape
   ```

Initiating a tape dump

You can accommodate a large dump by using multiple tapes. Only the first tape (that you IPL from) needs to have the tape dump tool installed.

To obtain a dump with the tape dump tool, perform the following main steps:
1. Set the cartridge loader to AUTO and insert a sufficient number of cartridges.
   **Attention:** The dump tool loads tapes automatically from the cartridge holder and overwrites any data on them. Be sure that the cartridge holder does not hold tapes with data that are still needed.
2. Ensure that the tapes are rewound.
3. Stop all CPUs.
4. Store status on the IPL CPU.
5. IPL the dump tool on the IPL CPU.

**Note:** Do not clear storage!

The dump tool writes messages to the tape drive message display (not to the operator console). First the number of dumped MB is displayed. When a tape cartridge is full, it is automatically unloaded and the message `next*vol` is displayed. If more cartridges are available, they are loaded and the dump continues. If no cartridge is available, the dump tool waits for the operator to load one.

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 should go 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
```

6. You can IPL Linux again.


---

**Tape display messages**

- `next*vol`
  - The dump tool loads the next tape cartridge, or, if none is available, waits for the operator to load one.

- `number`
  - The number of MB dumped.

- `dump*end`
  - The dump process ended successfully.

---

**Copying the dump from tape**

This section describes how to copy a tape dump to a file system using the `zgetdump` tool.

**Prerequisite:** You must have installed the `mt` utility.

---

**Preparing the dump tape**

You need to rewind the tape, and find the correct position on the tape to start copying from. Use the `mt` tool to do this.

1. Rewind the tape.
Example:

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

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

Example:

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

### Using the zgetdump tool

For a multivolume dump, the cartridges must be loaded in the right order, starting with the first volume. You can use the cartridge loader for automatic loading. When zgetdump finds the end of a volume (that is not the end of the dump) it writes a message and attempts to load the next cartridge.

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.

**Note:** Always use the AUTO setting on your tape device when working with zgetdump.

**Example:** Assuming that the tape is in the correct position (see "Preparing the dump tape" on page 12) and is on tape device /dev/ntibm0, use the following command to copy the dump from tape to a file dump_file in the file system:

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

For general information on zgetdump, see "The zgetdump tool" on page 29 or the man page.

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

To check whether a single-volume or a multivolume dump is valid, use the -i option. For multivolume dumps, use the -a option to signal that you are working with a multivolume dump. For example, to check whether a multivolume dump is valid:

1. Ensure that the volumes are loaded in the correct sequence.
2. Skip the first file on the first tape (this is the dump tool itself):

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

3. Issue:

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

zgetdump goes through all the volumes until it reaches the end of the dump. See also "Using zgetdump to copy a multi-volume tape dump" on page 31.
Chapter 5. Using a SCSI dump device

You can use SCSI disks that are accessed through the zfcp 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.

This section describes how to install the SCSI dump tool and how to initiate a SCSI dump.

Installing the SCSI disk dump tool

**Requirement:** The dump directory needs enough free space (memory size + 10 MB) to hold the system memory.

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 dump can be written to a second partition, the *dump partition*, provided it is on the same physical disk. Only the target partition need be mounted when zipl is run. In a single-partition configuration, the target partition is also the dump partition.

**SCSI dump tool parameters**

When installing the SCSI disk dump tool, the following parameters can be specified in a 'parameters' line in the zipl configuration file or using the '-P' option in the zipl command line.

**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.

**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_compress=gzip | none**

Dump compression option. Compression can be time-consuming on slower systems with a large amount of memory.

The default is “none”.

**dump_mode=interactive | auto**

Action taken if there is no room on the file system for the new dump file. "interactive" prompts the user to confirm that the 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 have been configured with zipl. For example, you might want to change the compression 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 operating system. For more information, see the SCSI examples in Appendix A, "Examples for initiating dumps," on page 21.
Example 1: Combined dump and target partition

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 also shows how to use the dump_compress parameter to generate the dump in gzip format.

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

   Example:
   ```
   # fdisk /dev/sda
   ```

   The created partition is /dev/sdal.

2. Format this partition with either the ext2 or ext3 file system.

   Example:
   ```
   # mke2fs -j /dev/sdal
   ```

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

   Example:
   ```
   # mount /dev/sdal /dumps
   # mkdir /dumps/mydumps
   ```

4. Install the dump tool using the zipl command. You can specify the dump device on the command line or use a configuration file.

   Command line example:
   ```
   # zipl -D /dev/sdal -t /dumps -P "dump_dir=/mydumps dump_compress=gzip"
   ```

   Configuration file example:
   a. Edit /etc/zipl.conf to add the following lines:

   ```
   [scsidump]
   target=/dumps
dumptofs=/dev/sdal
   parameters="dump_dir=/mydumps dump_compress=gzip"
   ```

   b. Issue zipl:

   ```
   # zipl scsidump
   ```

5. Unmount the file system:

   ```
   # umount /dumps
   ```

When you IPL /dev/sda1 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 4 on page 25 for an example.

Example 2: Menu configuration with separate dump and target partitions

This example assumes that a SCSI device /dev/sda is to be used as a dump device. In the example, the dump configuration is part of a menu configuration. Menu configurations are specified in a zipl configuration file and have a common
target directory that is specified in the menu section of the configuration file. To keep the dumps separated from other data, separate dump and target partitions are used. The example assumes that there are already three partitions:

- `/dev/sda1` is the production partition and mounted as the root file system.
- `/dev/sda2` is the target partition, has been formatted with the PC-BIOS disk layout, and is mounted under `/boot`. `/boot` contains files for two Linux boot configurations (parmfile, image-1, and image-2).
- `/dev/sda3` is the dump partition and has been formatted with the PC-BIOS disk layout. The dump files are to be written to the root directory of the dump partition.

1. On the dump partition, create either an ext2 or ext3 file system:

   ```
   # mke2fs -j /dev/sda3
   ```

2. Edit `/etc/zipl.conf` to add the following lines:

   ```
   [ipl1]
   image=/boot/image-1
   parmfile=/boot/parmfile
   target=/boot
   
   [ipl2]
   image=/boot/image-2
   parmfile=/boot/parmfile
   target=/boot
   
   [scsidump1]
   dumptofs=/dev/sda3
   parameters="dump_compress=gzip"
   target=/boot
   
   # Menu containing all 3 configurations
   :menu1
   1=ipl1
   2=ipl2
   3=scsidump1
   default=1
   target=/boot
   ```

3. Install the menu configuration, including the dump tool, by issuing:

   ```
   # zipl --menu menu1
   ```

When you specify the “scsidump1” configuration at IPL-time using boot program selector 3, the dump configuration is used and a system dump is initiated. The boot program selector is located on the load panel, see Figure 4 on page 25 for an example.

For more information on using a configuration file, see the `zipl.conf` man page or refer to Device Drivers, Features, and Commands on SUSE Linux Enterprise Server 11 SP1, SC34-2595.

### Initiating a SCSI dump

To initiate the dump, IPL the dump tool using the **SCSI dump** load type. See Appendix A, “Examples for initiating dumps,” on page 21.

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.
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 15).

**Example:** If there are already two dump files, “dump.0” and “dump.1”, in the dump directory, the new dump will be “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.

**Printing the dump header**

To print the dump file header use zgetdump with the 'i' option:

```
# zgetdump -i dump.0
> *zgetdump -i* checks if a dump on either
> a dasd volume or single tape is valid.
> If the tape is part of a multi-volume tape dump,
> it checks if it is a valid portion of the dump.
Dump device: dump.0
This is a lkcd dump:
Memory start : 0x0
Memory end : 0x40000000
Physical memory: 1073741824
Panic string : zSeries-dump (CPUID = ff00012320978000)
Number of pages: 262144
Page size : 4096
Magic number : 0xa8190173618f23ed
Version number : 8
```
Chapter 6. Using VMDUMP

Do not use VMDUMP to dump large 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.

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

VMDUMP does not need to be installed separately.

Intitiating a dump with VMDUMP

Issue the following command from the guest's 3270 console:

```
#CP VMDUMP
```

Result: VM stops the Linux guest and creates a dump file in the guest's VM reader. After the dump is complete the Linux guest continues operating.

You can use the “TO” option of the VMDUMP command to direct the dump to the reader of another guest of the same VM.

Example: To write the dump to a VM guest “linux02” issue:

```
#CP VMDUMP TO LINUX02
```

If you want to include NSSs and DCSSs in your dump, use the “ALL” operand:

```
#cp VMDUMP 0:ALL
```

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

Copying the dump to Linux

You can use the vmur command under Linux or the DUMPLOAD command under CMS to copy the dump file.

Using the vmur command

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 above 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:
The created file, myvmdump, can then be used as input to **crash**.

**Using the DUMPLOAD command**

Alternatively you can use the DUMPLOAD command under CMS to access the dump. The DUMPLOAD command copies the dump from the VM reader to the CMS file system.

From the CMS file system, you can then transfer the dump to a Linux file system, for example with **ftp**.

```bash
# 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%
```
Appendix A. Examples for initiating dumps

The following sections describe how to initiate a dump from different control points.

z/VM

The following examples assume the 64-bit mode. Corresponding 31-bit examples would have a different PSW but be the same otherwise.

Using DASD

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.

Using tape

If 193 is the tape device:

```
#cp rewind 193
#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.

Using SCSI

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

Assume your SCSI dump disk has the following parameters:
• WWPN: 4712076300ce93a7
• LUN: 4712000000000000
• FCP adapter device number: 4711
• Boot program selector: 3

To initiate the dump process, follow these steps:
# cp set dumpdev portname 47120763 00ce93a7 lun 47120000 00000000 bootprog 3
# cp ipl 4711 dump

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

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 compression 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.

**Using VMDUMP**

To initialize a dump with VMDUMP issue this command from your Linux guest’s 3270 console:

```
# cp vmdump
```

Dumping does not force you to IPL. If the Linux instance ran as required before dumping, it will continue running when the dump is completed.

**HMC or SE**

You can initiate a dump process on an LPAR from an HMC (Hardware Management Console) or SE (Support Element). The following description refers to an HMC, but the steps also apply to an SE.

The steps are similar for DASD, tape, and SCSI. Differences are noted where applicable. You cannot initiate a dump with VMDUMP from the HMC or SE.

To initiate the dump:

1. In the left 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 upper content area on the right.
2. Select the LPAR for which you want to initiate the dump.
3. In the **Tasks** area, expand **Recovery**. Proceed according to your dump device:
   - If you are dumping to DASD or tape, click **Stop all** in the **Recovery** list to stop all CPUs. Confirm when you are prompted to do so.
   - If you are dumping to a SCSI disk, skip this step and proceed with step 4 on page 23.
Figure 2 shows an example of an HMC with a selected mainframe system and LPAR. The Load and Stop all tasks can be seen in the expanded Recovery list.

4. Click Load in the Recovery list to display the Load panel.

For a dump to DASD or tape:
   a. Select Load type “Normal”.
   b. Select the Store status check box.
   c. Type the device number of the dump device into the Load address field.

Figure 3 on page 24 shows a Load panel with all entries and selections required to start the dump process for a DASD or tape dump device.
For a dump to SCSI disk:

a. Select Load type "SCSI dump".

b. Type the device number of the FCP adapter for the SCSI disk into the Load address field.

c. Type the World Wide Port name of the SCSI disk into the World wide port name field.

d. Type the Logical Unit Number of the SCSI disk into the Logical unit number field.

e. Type the configuration number of the dump IPL configuration 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. In Example 2: Menu configuration with separate dump and target partitions on page 16, the dump configuration has the number 3.

f. 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 compression 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 4 shows a Load panel with all entries and selections required to start the SCSI dump process.

Figure 4. Load panel with enabled SCSI feature for dumping to SCSI disk

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

When the dump has completed successfully, you can IPL Linux again.
Appendix B. Obtaining a dump with limited size

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

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 limit the dump size below the amount of memory actually used by the system to be dumped. Limiting the size to below 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/dasdcl,100M
```

An equivalent section in a configuration file could look like this:

```
[dump1]
dumpto=/dev/dasdcl,100M
```
Appendix C. Command summary

This chapter describes tools to work with dumps. The descriptions of the commands contain only the relevant options and parameters, for a full description refer to the man pages.

- The zgetdump tool
- The dumpconf tool
- The crash tool
- The vmconvert tool
- The vmur tool

The zgetdump tool

The zgetdump tool reads a dump from the given dump device and writes its contents to standard out, unless the operator redirects it to a file.

```
zgetdump Command
```

```
+----------------+-----------------------+-------------------+-------------------+
|                | zgetdump              | <dumpdevice>      | standard out      |
|                |                       | <dumpdevice>      | > <dump_file>     |
| -i             |                       | -a                |                   |
| -d             | -<dumpdevice>         | -h                |                   |
| -v             |                       |                   |                   |
```

Where:

- `<dumpdevice>` specifies the device or partition as follows:
  - For single-volume DASD: The partition containing the dump, for example `/dev/dasdb1`
  - For multi-volume DASD: The device node of one of the DASD devices containing the dump, for example `/dev/dasdb`
  - For tape: The device node of the tape containing the dump, for example `/dev/ntibm0`
  - For the `-d` option: The device node of the DASD device, for example `/dev/dasdb`

- `<dump_file>` is the file to which the output is redirected. The default is standard out.

- `-i` or `--info` displays the header information from the dump and performs a validity check.
- `-a` signals that the dump is a multi-volume tape dump.
- `-d` checks whether the specified ECKD or FBA device contains a valid dump record.
-h or --help
displays the help information for the command.

-v or --version
displays the version information for the command.

Examples

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 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 "Copying a multi-volume
dump to a file" on page 10.
Using zgetdump to copy a multi-volume tape dump

Assuming that the tape device is /dev/ntibm0:

```
# zgetdump /dev/ntibm0 > dump_file
Dump device: /dev/ntibm0

Tape Volume 0

>>> Dump header information <<<
Dump created on: Wed Jul 7 17:20:16 2004
Magic number: 0xa8190173618f23fd
Version number: 2
Header size: 4096
Page size: 4096
Physical memory: 268435456
Number of pages: 65536
cpu id: 0xff02453096720000
System Arch: s390 (ESA)
Build Arch: s390
>>> End of Dump header <<<

Reading dump content ...........................
End of Volume reached.
Waiting for next volume to be loaded... done

Tape Volume 1 of a multi volume dump.
Reading dump content ......
Dump End Marker found: this dump is valid.
```

Checking whether a multi-volume tape dump is valid, and printing the dump header

Assuming that the tape device is /dev/ntibm0:

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

"zgetdump -i -a" checks if a multi-volume tape dump is valid.
Please make sure that all volumes are loaded in sequence.

Dump device: /dev/ntibm0

Tape Volume 0

>> Dump header information <<<
...

>> End of Dump header <<<

Checking if the dump is valid - this might take a while...
Reached End of Volume 0.
Waiting for Volume 1 to be loaded... done

Tape Volume 1 of a multi volume dump.
Dump End found: This Dump is valid.
```
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
> zgetdump -i checks if a dump on either
> a dasd volume or single tape is valid.
> If the tape is part of a multi-volume tape dump,
> it checks if it is a valid portion of the dump.

Dump device: /dev/dasdb1

>>> Dump header information <<<
Dump created on: Mon Jul  5 16:53:40 2004
Magic number: 0xa8190173618f23fd
Version number: 2
Header size: 4096
Page size: 4096
Physical memory: 268435456
Number of pages: 65536
cpu id: 0xff20000620640000
System Arch: s390 (ESA)
Build Arch: s390x (ESAME)
>>> End of Dump header <<<

Dump ended on: Mon Jul  5 16:54:06 2004
Dump End Marker found: this dump is valid.
```

Checking whether a device contains a valid dump record

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

```
# zgetdump -d /dev/dasda
'/dev/dasda' is Version 1 s390x (ESAME) dump device.
```

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

```
# zgetdump -d /dev/dasdc
'/dev/dasdc' is no dump device.
```

The dumpconf tool

The `dumpconf` tool configures the action to be taken if a kernel panic occurs. The command is installed as a service script under `/etc/init.d/dumpconf` and reads the configuration file `/etc/sysconfig/dumpconf`.

To enable the `dumpconf` service, issue:

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

**Before you start:** You need root permissions.
dumpconf syntax

```
dumpconf
  start
  stop
  status
```

Where:
- **start**  Enable configuration defined in `/etc/sysconfig/dumpconf`
- **stop**   Disable dumpconf.
- **status** Show current configuration status of dumpconf.

- **-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 occurs. Possible values are:
  - **dump**  Dump Linux and stop system.
  - **reipl**  Reboot Linux.
  - **dump_reipl**  Dump Linux and reboot system. Note that `dump_reipl` is only available on LPAR with z9® machines and later, and on z/VM with version 5.3 and later.
  - **vmcmd**  Execute specified CP commands and stop system.
  - **stop**  Stop Linux (default).

**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. Note that VM commands, device adresses, and VM guest names must be uppercase.

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

**DEVICE**
- Device number of 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.
Examples

Example configuration files for dumpconf:

- Example configuration for CCW dump device (DASD) and reipl after dump:

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

- 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 occurs:

```
ON_PANIC=reipl
```

- Example of sending a message to guest "MASTER", executing a CP VMDUMP command, and rebooting from device 4711 if a kernel panic occurs:

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

Note that VM commands, device adresses, and VM guest names must be uppercase.

Examples of dumpconf use: Use dumpconf 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
  type....: no dump device configured
  on_panic: stop
```
The crash tool

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

The vmconvert tool

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

```
vmconvert syntax

```

Where:

* `<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.
```
The vmur tool

The vmur command can receive a VMDUMP file from the 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>
```

Where:

- `<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, SC33-8411.

Example

To receive and convert a VMDUMP spool file with spool ID 463 to a file called `dump_file` on the Linux file system in the current working directory:

```
# vmur rec -c 463 dump_file
```
Appendix D. 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.

If the dump is to be analyzed with crash, include:
- vmlinux (text): Contains addresses of kernel symbols
- vmlinux (debug): Contains datatype debug information

SLES debug files

The SLES debug files are:

Table 2. SUSE Linux Enterprise Server debug file names

<table>
<thead>
<tr>
<th>Debug file</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>System.map</td>
<td>/boot/System.map-2.6.&lt;xx.yy-zz&gt;-default</td>
</tr>
<tr>
<td>Kerntypes</td>
<td>/boot/Kerntypes-2.6.&lt;xx.yy-zz&gt;-default</td>
</tr>
<tr>
<td>vmlinux (text)</td>
<td>/boot/vmlinux-2.6.&lt;xx.yy-zz&gt;-default.gz</td>
</tr>
<tr>
<td>vmlinux (debug)</td>
<td>/usr/lib/debug/boot/vmlinux-2.6.&lt;xx.yy-zz&gt;-default.debug</td>
</tr>
</tbody>
</table>

The RPMs that contain the files are:

Table 3. SUSE Linux Enterprise Server RPM names

<table>
<thead>
<tr>
<th>SLES version</th>
<th>RPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLES 11</td>
<td>• kernel-default-base-2.6.&lt;xx.yy-zz&gt;.rpm</td>
</tr>
<tr>
<td></td>
<td>• kernel-default-debuginfo-2.6.&lt;xx.yy-zz&gt;.rpm</td>
</tr>
</tbody>
</table>
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 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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Readers’ Comments — We'd Like to Hear from You

Linux on System z
Using the Dump Tools
on SUSE Linux Enterprise Server 11 SP1

Publication No. SC34-2598-00

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 e-mail to: eservdoc@de.ibm.com

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

Name

Address

Company or Organization

Phone No.

E-mail address