

**NAME**

**binmiscctl** - manage binary image activators

**SYNOPSIS**

```
binmiscctl add name --interpreter path --magic magic --size size [--mask mask] [--offset offset]
[--set-enabled] [--pre-open]
binmiscctl disable name
binmiscctl enable name
binmiscctl list
binmiscctl lookup name
binmiscctl remove name
```

**DESCRIPTION**

The **binmiscctl** utility is the management utility for configuring miscellaneous binaries image activators in the kernel. It allows adding, deleting, disabling, enabling, and looking up interpreters. Also, all the interpreters can be listed.

The first argument on the command line indicates the operation to be performed. Operation must be one of the following:

```
add name --interpreter path --magic magic --size size [--mask mask] [--offset offset] [--set-enabled]
[--pre-open]
```

Add a new activator entry in the kernel. You must specify a unique *name*, a *path* to the interpreter, header *magic* bytes that uniquely identify a suitable binary for the activator, and the *size* of the *magic* in bytes.

Optionally, you may specify a *mask* to do a bitwise AND with the header bytes. This effectively allows you to ignore fields in the binary header that do not uniquely identify the binary file's type.

An *offset* may be specified for the magic bytes using the **--offset** option. By default the *offset* is zero.

To enable the activator entry the **--set-enabled** option is used. The activator default state is disabled.

To make the interpreter automatically available in jails and chroots, use the **--pre-open** option to allow the kernel to open the binary at configuration time rather than lazily when the the interpreted program is started.

The interpreter *path* may also contain arguments for the interpreter including *#a* which gets replaced by the old `argv0` value in the interpreter string.

**disable** *name*

Disable the activator entry identified with *name*.

**enable** *name*

Enable the activator entry identified with *name*.

**list** Take a snapshot and print all the activator entries currently configured.

**lookup** *name*

Look up and print out the activator entry identified with *name*.

**remove** *name*

Remove the activator entry identified with *name*.

**EXAMPLES**

Add an image activator to run the LLVM interpreter (`lli(1)`) on bitcode compiled files and set its state to enabled. In this example *#a* is replaced with the old `argv0` value so that `lli(1)` can fake its `argv0`:

```
# binmiscctl add llvmbc --interpreter '/usr/bin/lli \  
--fake-argv0=#a' --magic 'BC\x0\xde' --size 4 \  
--set-enabled
```

Set the state of the *llvmbc* image activator to disabled:

```
# binmiscctl disable llvmbc
```

Set the state of the *llvmbc* image activator to enabled:

```
# binmiscctl enable llvmbc
```

Delete the *llvmbc* image activator:

```
# binmiscctl remove llvmbc
```

Look up and list the record for the *llvmbc* image activator:

```
# binmiscctl lookup llvmbc
```

Add QEMU `bsd-user` program as an image activator for ARM AARCH64 binaries:

```
# binmiscctl add arm64 \
--interpreter "/usr/local/bin/qemu-aarch64-static" \
--magic "\x7f\x45\x4c\x46\x02\x01\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x02\x00\xb7\x00" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

Add QEMU `bsd-user` program as an image activator for ARM little-endian binaries:

```
# binmiscctl add armelf \
--interpreter "/usr/local/bin/qemu-arm-static" \
--magic "\x7f\x45\x4c\x46\x01\x01\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x02\x00x28\x00" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

Add QEMU `bsd-user` program as an image activator for ARM big-endian binaries:

```
# binmiscctl add armebelf \
--interpreter "/usr/local/bin/qemu-arm-static" \
--magic "\x7f\x45\x4c\x46\x01\x02\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x02\x00x28" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

Add QEMU `bsd-user` program as an image activator for MIPS32 binaries:

```
# binmiscctl add mips32 \
--interpreter "/usr/local/bin/qemu-mips-static" \
--magic "\x7f\x45\x4c\x46\x01\x02\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x02\x00x08" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

Add QEMU bsd-user program as an image activator for MIPS64 binaries:

```
# binmiscctl add mips64 \
--interpreter "/usr/local/bin/qemu-mips64-static" \
--magic "\x7f\x45\x4c\x46\x02\x02\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x00\x02\x00\x08" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

Add QEMU bsd-user program as an image activator for PowerPC binaries:

```
# binmiscctl add powerpc \
--interpreter "/usr/local/bin/qemu-ppc-static" \
--magic "\x7f\x45\x4c\x46\x01\x02\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x00\x02\x00\x14" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

Add QEMU bsd-user program as an image activator for PowerPC64 binaries:

```
# binmiscctl add powerpc64 \
--interpreter "/usr/local/bin/qemu-ppc64-static" \
--magic "\x7f\x45\x4c\x46\x02\x02\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x00\x02\x00\x15" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

Add QEMU bsd-user program as an image activator for 64-bit RISC-V binaries:

```
# binmiscctl add riscv64 \
--interpreter "/usr/local/bin/qemu-riscv64-static" \
--magic "\x7f\x45\x4c\x46\x02\x01\x01\x00\x00\x00\
\x00\x00\x00\x00\x00\x00\x02\x00\xf3\x00" \
--mask "\xff\xff\xff\xff\xff\xff\xff\x00\xff\xff\
\xff\xff\xff\xff\xff\xff\xff\xfe\xff\xff" \
--size 20 --set-enabled
```

### Create and use an ARMv6 chroot on an AMD64 host

Use an existing source tree to build a chroot host with architecture overrides:

```
D=/path/to/chroot
cd /usr/src
mkdir -p $D
make world TARGET=arm TARGET_ARCH=armv6 DESTDIR=$D
make distribution TARGET=arm TARGET_ARCH=armv6 DESTDIR=$D
```

With *emulators/qemu-user-static* from the FreeBSD Ports Collection, the emulator must be copied into the jail path specified in the **binmiscctl** command. Using the example above:

```
mkdir $D/usr/local/bin
cp /usr/local/bin/qemu-arm-static $D/usr/local/bin
```

Now the user can chroot into the environment normally, as root:

```
chroot $D
```

### SEE ALSO

lli(1), execve(2), jail(8)

### HISTORY

The **binmiscctl** command was added in FreeBSD 10.1. It was developed to support the `imgact_binmisc` kernel module.

### AUTHORS

Stacey D Son <[sson@FreeBSD.org](mailto:sson@FreeBSD.org)>