

NAME

release - release building infrastructure

DESCRIPTION

FreeBSD provides a complete build environment suitable for users to make full releases of the FreeBSD operating system. All of the tools necessary to build a release are available from the FreeBSD source code repository in *src/release*. A complete release can be built with only a single command, including the creation of ISO images suitable for burning to CD-ROM, memory stick images, and a network install directory. This command is aptly named "make release".

For some users, it may be desirable to provide an absolutely clean build environment, with no local modifications to the source tree or to *make.conf*(5), and with clean checkouts of specific versions of the *doc*, *src*, and *ports* trees. For this purpose, a script (*src/release/release.sh*) is provided to automate these checkouts and then execute "make release" in a clean chroot(8).

Before attempting to build a release, the user is expected to be familiar with the contents of *build*(7), and should have experience upgrading systems from source.

The release build process requires that */usr/obj* be populated with the output of "make buildworld" and "make buildkernel". This is necessary to provide the object files for the release or, when using *release.sh*, so that the object files for a complete system can be installed into a clean chroot(8) environment.

If the target release build is for a different architecture or machine type, the *TARGET* and *TARGET_ARCH* variables must be used. See the supported *release.conf* variables for more information.

The release procedure on some architectures may also require that the *md*(4) (memory disk) device driver be present in the kernel (either by being compiled in or available as a module).

This document does not cover source code management, quality assurance, or other aspects of the release engineering process.

CLEAN RELEASE GENERATION

Official releases of FreeBSD are produced in a clean environment to ensure consistency between the versions of the *src*, *ports*, and *doc* trees and to avoid contamination from the host system (such as local patches, changes to *make.conf*(5), etc.). This is accomplished using the wrapper script *src/release/release.sh*.

release.sh [-c *release.conf*]

release.sh checks out the `src/`, `ports/`, and `doc/` trees to `CHROOTDIR`, then calls "make buildworld" and "make installworld" to generate a chroot(8) environment. Next, "make release" is run within the chroot(8) environment and places the result in `$CHROOTDIR/R`.

The optional `release.conf` configuration file supports the following variables:

CHROOTDIR The directory within which the release will be built.

CHROOT_MAKEENV

Additional make(1) arguments to pass through, which directly affect the tuning of the build chroot.

NOGIT Do not explicitly require the git(1) port to be installed.

GITROOT The git(1) host used to check out the various trees. Defaults to `https://git.FreeBSD.org`.

SRCBRANCH The `src/` branch to use. Defaults to `-b main`.

PORTBRANCH The `ports/` branch to use. Defaults to `head/@rHEAD`.

TARGET The target machine type for cross-building a release.

TARGET_ARCH The target machine architecture for cross-building a release.

For the supported list of `TARGET` and `TARGET_ARCH` combinations, consult the output of "make targets" as documented in build(7).

KERNEL The target kernel configuration to use. Defaults to `GENERIC`. Multiple `KERNEL` entries may be specified.

MAKE_CONF The `make.conf(5)` to use for the release build. Defaults to `/dev/null` to prevent polluting the release with local system changes.

SRC_CONF The `src.conf(5)` to use for the release build. Defaults to `/dev/null` to prevent polluting the release with local system changes.

MAKE_FLAGS Additional flags to pass to make(1).

WORLD_FLAGS Additional flags to pass to make(1) during the "buildworld" phase. Defaults to

setting the number of `make(1)` jobs (`-j`) to the number of CPUs available on a SMP-capable system.

KERNEL_FLAGS

Additional flags to pass to `make(1)` during the "buildkernel" phase. Defaults to setting the number of `make(1)` jobs (`-j`) to half the number of CPUs available on a SMP-capable system.

NOPORTS

Set to a non-empty value to skip the `ports/` tree checkout. When set, *NOPORTS* will prevent the *ports.txz* distribution package from being created.

WITH_DVD

Set to a non-empty value to include the **dvdrom** target.

WITH_COMPRESSED_IMAGES

Set to a non-empty value to compress the release images with `xz(1)`. The original (uncompressed) images are not removed.

XZ_THREADS (*int*)

Set to the number of threads `xz(1)` should use when compressing images. By default, *XZ_THREADS* is set to 0, which uses all available cores on the system.

VCSCMD

The command run to obtain the source trees. Defaults to "**git clone -q**".

CHROOTBUILD_SKIP

If defined, the `buildworld`, `installworld`, and `distribution` stages of the `chroot(8)` build environment setup are skipped. This is intended solely for cases where the `chroot(8)` userland are provided by alternate means.

SRC_UPDATE_SKIP

Set to a non-empty value to prevent checkout or update of `/usr/src` within the `chroot(8)`. This is intended for use only when `/usr/src` is expected to exist by alternative means.

PORTS_UPDATE_SKIP

Set to a non-empty value to prevent checkout or update of `/usr/ports` within the `chroot(8)`. This is intended for use only when `/usr/ports` is expected to exist by alternative means.

EMBEDDED BUILDS

The following *release.conf* variables are relevant only to release builds for embedded systems:

EMBEDDEDBUILD

Set to a non-null value to enable functionality for embedded device release builds.

When set, *WITH_DVD* is unset. Additionally, *EMBEDDED_TARGET* and *EMBEDDED_TARGET_ARCH* must also be defined. When the build environment is created, *release.sh* runs a separate build script located in an architecture-specific directory in *src/release/\${EMBEDDED_TARGET}/*.

EMBEDDEDPORTS

Set to the list of any ports that are required for the target device in the format of *category/port*.

EMBEDDED_TARGET

When set, its value is passed to *make(1)* to set the *TARGET* (value of **uname -m**) to cross build the target userland.

EMBEDDED_TARGET_ARCH

When set, its value is passed to *make(1)* to set the *TARGET_ARCH* (value of **uname -p**) to cross build the target userland.

VIRTUAL MACHINE DISK IMAGES

The following *release.conf* variables are relevant only to virtual machine disk image builds:

WITH_VMIMAGES

Set to a non-null value to build virtual machine disk images as part of the release build. *WITH_VMIMAGES* may also be specified as an environment variable passed to *make(1)*.

WITH_COMPRESSED_VMIMAGES

Set to a non-null value to compress the virtual machine disk images with *xz(1)* as part of the **install** *make(1)* target. Note that compressing virtual machine disk images may take a very long time on some systems.

VMBASE

Set to change the name of the resulting virtual machine disk image file. The default value is *vm*.

VMSIZE

Set to change the size of the virtual machine disk capacity. The default value is *20g*. See *makefs(8)* for valid values.

Virtual machine disk images are, by default, created as sparse images. When

WITH_COMPRESSED_VMIMAGES is used, the resulting files compressed with *xz(1)* compress to roughly the same size, regardless of the specified disk image size.

VMFS (Deprecated.) Set to specify which of the filesystem(s) listed in *VMFSLIST* is linked to the historical non-filesystem-labelled file name. Valid values are *ufs* and *zfs*. The default value is *ufs*.

VMFSLIST Set to specify the list of file system types to build images for. Valid values are one or both of *ufs* and *zfs*. The default value is *ufs zfs*.

VMFORMATS Set to the target virtual disk image format(s) to create. By default, the *vhdf*, *vmrk*, *qcow2*, and *raw* formats are created. See *mkimg(1)* for valid format values.

For a list of supported *VMFORMATS* values (including cloud hosting provider formats) along with a brief description, run:

```
cd /usr/src
make -C release list-vmtargets
```

CLOUD HOSTING MACHINE IMAGES

The FreeBSD release build tools support building virtual machine images for various cloud hosting providers, each with their own specific configuration to include support for each hosting provider by default.

The following *make(1)* environment variables are supported:

CLOUDWARE Set to a list of one or more cloud hosting providers, enclosed in quotes. Requires *WITH_CLOUDWARE* to also be set.

WITH_CLOUDWARE Set to a non-empty value to enable building virtual machine images for various cloud hosting providers. Requires *CLOUDWARE* to also be set.

Additionally, the *CLOUDWARE* and *WITH_CLOUDWARE* variables can be added to *release.conf*, and used in conjunction with *release.sh*.

For a list of supported *CLOUDWARE* values, run:

```
cd /usr/src
make -C release list-cloudware
```

MAKEFILE TARGETS

The release makefile (*src/release/Makefile*) is fairly abstruse. Most developers will only be concerned with the **release** and **install** targets.

- release** Meta-target to build all release media and distributions applicable to this platform.
- install** Copy all produced release media to *\${DESTDIR}*.
- cdrom** Builds installation CD-ROM images. This may require the md(4) (memory disk) device driver be present in the kernel (either by being compiled in or available as a module). This target produces files called *disc1.iso* and *bootonly.iso* as its output.
- dvdrom** Builds installation DVD-ROM images. This may require the md(4) (memory disk) device driver be present in the kernel (either by being compiled in or available as a module). This target produces the *dvd1.iso* file as its output.
- memstick** Builds an installation memory stick image named *memstick.img*. Not applicable on all platforms. Requires that the md(4) (memory disk) device driver be present in the kernel (either by being compiled in or available as a module).
- mini-memstick** Similar to **memstick**, with the exception that the installation distribution sets are not included.
- ftp** Creates a directory named *ftp* containing the distribution files used in network installations and suitable for upload to an FTP mirror.
- vm-image** Creates virtual machine disk images in various formats. The **vm-image** target requires the *WITH_VMIMAGES* make(1) environment variable to be set to a non-null value.
- vm-cloudware** Builds FreeBSD virtual machine images for various cloud hosting providers. See "CLOUD HOSTING MACHINE IMAGES" for implementation details.
- list-cloudware** Displays the list of valid *CLOUDWARE* values.
- list-vmtargets** Displays the list of valid *VMFORMATS* and *CLOUDWARE* values.
- Major subtargets called by targets above:
- packagesystem** Generates all the distribution archives (base, kernel, ports, doc) applicable on this

platform.

- disc1** Builds a bootable installation system containing all the distribution files packaged by the **packagesystem** target, and suitable for imaging by the **cdrom**, **dvdrom** and **memstick** targets.
- reldoc** Builds the release documentation. This includes the release notes, hardware guide, and installation instructions. Other documentation, such as the Handbook, is built during the **base.txz** target invoked by **packagesystem**.

ENVIRONMENT

Optional variables:

- OSRELEASE** Optional base name for generated media images when invoking the **install** target (e.g., FreeBSD-12.1-RELEASE-amd64). Defaults to the output of **'uname -s'-'uname -r'-'uname -p'** within the chroot.
- WORLDDIR** Location of a directory containing the src tree. By default, the directory above the one containing the makefile (*src*).
- PORTSDIR** Location of a directory containing the ports tree. By default, */usr/ports*. If it is unset or cannot be found, ports will not be included in the release.
- NOPORTS** If defined, the Ports Collection will be omitted from the release.
- NOSRC** If set, do not include system source code in the release.
- TARGET** The target hardware platform. This is analogous to the **"uname -m"** output. This is necessary to cross-build some target architectures. For example, cross-building for ARM64 machines requires **TARGET_ARCH=aarch64** and **TARGET=arm64**. If not set, **TARGET** defaults to the current hardware platform.
- TARGET_ARCH** The target machine processor architecture. This is analogous to the **"uname -p"** output. Set this to cross-build for a different architecture. If not set, **TARGET_ARCH** defaults to the current machine architecture, unless **TARGET** is also set, in which case it defaults to the appropriate value for that platform. Typically, one only needs to set **TARGET**.

FILES

/usr/doc/Makefile

```
/usr/doc/share/mk/doc.project.mk  
/usr/ports/Mk/bsd.port.mk  
/usr/ports/Mk/bsd.sites.mk  
/usr/share/examples/etc/make.conf  
/usr/src/Makefile  
/usr/src/Makefile.inc1  
/usr/src/release/Makefile  
/usr/src/release/Makefile.vm  
/usr/src/release/release.sh  
/usr/src/release/release.conf.sample  
/usr/src/release/tools/*.conf  
/usr/src/release/tools/vmimage.subr
```

EXAMPLES

The following sequence of commands can be used to build a "-CURRENT snapshot":

```
cd /usr  
git clone -b main https://git.freebsd.org/src.git src  
cd src  
make buildworld buildkernel  
cd release  
make obj  
make release  
make install DESTDIR=/var/freebsd-snapshot
```

After running these commands, all produced distribution files (tarballs for FTP, CD-ROM images, etc.) are available in the */var/freebsd-snapshot* directory.

The following sequence of commands can be used to build a "-CURRENT snapshot" in a clean environment, including ports and documentation:

```
cd /usr/src/release  
sh release.sh
```

Optionally, a configuration file can be used to customize the release build:

```
cd /usr/src/release  
sh release.sh -c $HOME/release.conf
```

Configuration files specific to various supported embedded systems, such as the Raspberry Pi, exist in

the directory corresponding to the *TARGET* `make(1)` variable. For example, to build an image for the Raspberry Pi:

```
cd /usr/src/release
sh release.sh -c arm/RPI-B.conf
```

To build an image for the Raspberry Pi 3:

```
cd /usr/src/release
sh release.sh -c arm64/RPI3.conf
```

After running these commands, all prepared release files are available in the */scratch* directory. The target directory can be changed by specifying the *CHROOTDIR* variable in *release.conf*.

COMPATIBILITY

The `reldoc` target was removed in commit `f61e92ca5a23`, and `DOCDIR`, `DOCBRANCH`, `DOC_UPDATE_SKIP`, and `NODOC` are therefore no longer supported.

SEE ALSO

`cc(1)`, `git(1)` (*ports/devel/git*), `install(1)`, `make(1)`, `mkimg(1)`, `uname(1)`, `md(4)`, `make.conf(5)`, `build(7)`, `ports(7)`, `chroot(8)`, `mtree(8)`, `sysctl(8)`

FreeBSD Release Engineering, <https://docs.freebsd.org/en/articles/freebsd-releng/>.

FreeBSD Developers' Handbook, <https://docs.freebsd.org/en/books/developers-handbook/>.

HISTORY

FreeBSD 1.x used a manual checklist, compiled by Rod Grimes, to produce a release. Apart from being incomplete, the list put a lot of specific demands on available file systems and was quite torturous to execute.

As part of the FreeBSD 2.0 release engineering effort, significant effort was spent getting *src/release/Makefile* into a shape where it could at least automate most of the tediousness of building a release in a sterile environment.

For the FreeBSD 9.0 release, *src/release/Makefile* was overhauled and the wrapper script *src/release/generate-release.sh* introduced to support the introduction of a new installer.

For the FreeBSD 9.2 release, *src/release/release.sh* was introduced to support per-build configuration files. *src/release/release.sh* is heavily based on the *src/release/generate-release.sh* script.

At near 1000 revisions spread over multiple branches, the `git(1)` log of `src/release/Makefile` contains a vivid historical record of some of the hardships release engineers go through.

AUTHORS

`src/release/Makefile` was originally written by Rod Grimes, Jordan Hubbard, and Poul-Henning Kamp.

This manual page was originally written by Murray Stokely <murray@FreeBSD.org>.

It was updated by Nathan Whitehorn <nwhitehorn@FreeBSD.org> to include the `generate-release.sh` script used for the FreeBSD 9.0 release cycle.

It was later updated by Glen Barber <gjb@FreeBSD.org> to include the `release.sh` script used for the FreeBSD 9.2 release cycle.