Build Your Own FreeBSD Update Server

The instructions in this article refer to an older version of FreeBSD and may not work properly on recent versions of the OS. With the availability of pkgbase, the freebsd-update utility is scheduled to be removed from FreeBSD in the future. When that happens, this article is either updated to reflect the new procedures or removed entirely.

Abstract

This article describes building an internal FreeBSD Update Server. The freebsd-update-server is written by Colin Percival <cperciva@FreeBSD.org>, Security Officer Emeritus of FreeBSD. For users that think it is convenient to update their systems against an official update server, building their own FreeBSD Update Server may help to extend its functionality by supporting manually-tweaked FreeBSD releases or by providing a local mirror that will allow faster updates for a number of machines.

Table of Contents

1. Acknowledgments 1
2. Introduction 1
3. Prerequisites 2
4. Configuration: Installation & Setup 2
5. Building Update Code 4
6. Building a Patch 8
7. Tips 12

1. Acknowledgments

This article was subsequently printed at BSD Magazine.

2. Introduction

Experienced users or administrators are often responsible for several machines or environments. They understand the difficult demands and challenges of maintaining such an infrastructure. Running a FreeBSD Update Server makes it easier to deploy security and software patches to selected test machines before rolling them out to production. It also means a number of systems can be updated from the local network rather than a potentially slower Internet connection. This article outlines the steps involved in creating an internal FreeBSD Update Server.

---

3. Prerequisites

To build an internal FreeBSD Update Server some requirements should be met.

- A running FreeBSD system.

At a minimum, updates require building on a FreeBSD release greater than or equal to the target release version for distribution.

- A user account with at least 4 GB of available space. This will allow the creation of updates for 7.1 and 7.2, but the exact space requirements may change from version to version.
- An ssh(1) account on a remote machine to upload distributed updates.
- A web server, like Apache, with over half of the space required for the build. For instance, test builds for 7.1 and 7.2 consume a total amount of 4 GB, and the webserver space needed to distribute these updates is 2.6 GB.
- Basic knowledge of shell scripting with Bourne shell, sh(1).

4. Configuration: Installation & Setup

Download the freebsd-update-server software by installing devel/git and security/ca_root_nss, and execute:

```
% git clone https://github.com/freebsd/freebsd-update-build.git freebsd-update-server
```

Update scripts/build.conf appropriately. It is sourced during all build operations.

Here is the default build.conf, which should be modified to suit your environment.

```
# Main configuration file for FreeBSD Update builds.  The
# release-specific configuration data is lower down in
# the scripts tree.

# Location from which to fetch releases
export FTP=ftp://ftp2.freebsd.org/pub/FreeBSD/releases ①

# Host platform
export HOSTPLATFORM=`uname -m`

# Host name to use inside jails
export BUILDHOSTNAME=${HOSTPLATFORM}-builder.daemonology.net ②

# Location of SSH key
export SSHKEY=/root/.ssh/id_dsa ③

# SSH account into which files are uploaded
MASTERACCT=builder@wadham.daemonology.net ④
```

```
# Directory into which files are uploaded
MASTERDIR=update-master.freebsd.org ⑤
```

Parameters for consideration would be:

① This is the location where ISO images are downloaded from (by the `fetchiso()` subroutine of scripts/build.subr). The location configured is not limited to FTP URIs. Any URI scheme supported by standard fetch(1) utility should work fine. Customizations to the `fetchiso()` code can be installed by copying the default build.subr script to the release and architecture-specific area at scripts/RELEASE/ARCHITECTURE/build.subr and applying local changes.

② The name of the build host. This information will be displayed on updated systems when issuing:

```
% uname -v
```

③ The SSH key for uploading files to the update server. A key pair can be created by typing `ssh-keygen -t dsa`. This parameter is optional; standard password authentication will be used as a fallback authentication method when `SSHKEY` is not defined. The ssh-keygen(1) manual page has more detailed information about SSH and the appropriate steps for creating and using one.

④ Account for uploading files to the update server.

⑤ Directory on the update server where files are uploaded to.

The default build.conf shipped with the freebsd-update-server sources is suitable for building i386 releases of FreeBSD. As an example of building an update server for other architectures, the following steps outline the configuration changes needed for amd64:

1. Create a build environment for amd64:

   ```
   % mkdir -p /usr/local/freebsd-update-server/scripts/7.2-RELEASE/amd64
   ```

2. Install a build.conf in the newly created build directory. The build configuration options for FreeBSD 7.2-RELEASE on amd64 should be similar to:

   ```
   # SHA256 hash of RELEASE disc1.iso image.
   export RELH=1ea1f6f652d7c5f5eab7ef9f8edbed50cb664b08ed761850f95f48e86cc71ef5 ①
   # Components of the world, source, and kernels
   export WORLDPARTS="base catpages dict doc games info manpages proflibs lib32"
   export SOURCEPARTS="base bin contrib crypto etc games gnu include krb5  \
                   lib libexec release rescue sbin secure share sys tools  \
                   ubin usbin cddl"
   export KERNELPARTS="generic"

   # EOL date
   ```

```
export EOL=1275289200 ②
```

① The sha256(1) hash key for the desired release, is published within the respective release announcement.

② To generate the "End of Life" number for build.conf, refer to the "Estimated EOL" posted on the FreeBSD Security Website. The value of EOL can be derived from the date listed on the web site, using the date(1) utility, for example:

```
% date -j -f '%Y%m%d-%H%M%S' '20090401-000000' '+%s'
```

5. Building Update Code

The first step is to run scripts/make.sh. This will build some binaries, create directories, and generate an RSA signing key used for approving builds. In this step, a passphrase will have to be supplied for the final creation of the signing key.

```
# sh scripts/make.sh
cc -O2 -fno-strict-aliasing -pipe   findstamps.c  -o findstamps
findstamps.c: In function 'usage':
findstamps.c:45: warning: incompatible implicit declaration of built-in function
'exit'
cc -O2 -fno-strict-aliasing -pipe   unstamp.c  -o unstamp
install findstamps ../bin
install unstamp ../bin
rm -f findstamps unstamp
Generating RSA private key, 4096 bit long modulus
.....................................................................................++
.....................++
e is 65537 (0x10001)

Public key fingerprint:
27ef53e48dc869eea6c3136091cc6ab8589f967559824779e855d58a2294de9e

Encrypting signing key for root
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
```

Keep a note of the generated key fingerprint. This value is required in /etc/freebsd-update.conf for binary updates.

At this point, we are ready to stage a build.

```
# cd /usr/local/freebsd-update-server
# sh scripts/init.sh amd64 7.2-RELEASE
```

What follows is a sample of an *initial* build run.

```
# sh scripts/init.sh amd64 7.2-RELEASE
Mon Aug 24 16:04:36 PDT 2009 Starting fetch for FreeBSD/amd64 7.2-RELEASE
/usr/local/freebsd-update-server/work/7.2-RELE100 of  588 MB  359 kBps 00m00s
Mon Aug 24 16:32:38 PDT 2009 Verifying disc1 hash for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 16:32:44 PDT 2009 Extracting components for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 16:34:05 PDT 2009 Constructing world+src image for FreeBSD/amd64 7.2-
RELEASE
Mon Aug 24 16:35:57 PDT 2009 Extracting world+src for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 23:36:24 UTC 2009 Building world for FreeBSD/amd64 7.2-RELEASE
Tue Aug 25 00:31:29 UTC 2009 Distributing world for FreeBSD/amd64 7.2-RELEASE
Tue Aug 25 00:32:36 UTC 2009 Building and distributing kernels for FreeBSD/amd64 7.2-
RELEASE
Tue Aug 25 00:44:44 UTC 2009 Constructing world components for FreeBSD/amd64 7.2-
RELEASE
Tue Aug 25 00:44:56 UTC 2009 Distributing source for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 17:46:18 PDT 2009 Moving components into staging area for FreeBSD/amd64
7.2-RELEASE
Mon Aug 24 17:46:33 PDT 2009 Identifying extra documentation for FreeBSD/amd64 7.2-
RELEASE
Mon Aug 24 17:47:13 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 17:47:18 PDT 2009 Indexing release for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 17:50:44 PDT 2009 Indexing world0 for FreeBSD/amd64 7.2-RELEASE

Files built but not released:
Files released but not built:
Files which differ by more than contents:
Files which differ between release and build:
kernel|generic|/GENERIC/hptrr.ko
kernel|generic|/GENERIC/kernel
src|sys|/sys/conf/newvers.sh
world|base|/boot/loader
world|base|/boot/pxeboot
world|base|/etc/mail/freebsd.cf
world|base|/etc/mail/freebsd.submit.cf
world|base|/etc/mail/sendmail.cf
world|base|/etc/mail/submit.cf
world|base|/lib/libcrypto.so.5
world|base|/usr/bin/ntpq
world|base|/usr/lib/libalias.a
world|base|/usr/lib/libalias_cuseeme.a
world|base|/usr/lib/libalias_dummy.a
world|base|/usr/lib/libalias_ftp.a
...
```

Then the build of the world is performed again, with world patches. A more detailed explanation may be found in scripts/build.subr.

During this second build cycle, the network time protocol daemon, ntpd(8), is turned off. Per Colin Percival <cperciva@FreeBSD.org>, Security Officer Emeritus of FreeBSD, "the freebsd-update-server build code needs to identify timestamps which are stored in files so that they can be ignored when comparing builds to determine which files need to be updated. This timestamp-finding works by doing two builds 400 days apart and comparing the results."

```
Mon Aug 24 17:54:07 PDT 2009 Extracting world+src for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 00:54:34 UTC 2010 Building world for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 01:49:42 UTC 2010 Distributing world for FreeBSD/amd64 7.2-RELEASE
Wed Sep 29 01:50:50 UTC 2010 Building and distributing kernels for FreeBSD/amd64 7.2-
RELEASE
Wed Sep 29 02:02:56 UTC 2010 Constructing world components for FreeBSD/amd64 7.2-
RELEASE
Wed Sep 29 02:03:08 UTC 2010 Distributing source for FreeBSD/amd64 7.2-RELEASE
Tue Sep 28 19:04:31 PDT 2010 Moving components into staging area for FreeBSD/amd64
7.2-RELEASE
Mon Aug 24 19:04:46 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:04:51 PDT 2009 Indexing world1 for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:08:04 PDT 2009 Locating build stamps for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:10:19 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:10:19 PDT 2009 Preparing to copy files into staging area for
FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 19:10:20 PDT 2009 Copying data files into staging area for FreeBSD/amd64
7.2-RELEASE
Mon Aug 24 12:16:57 PDT 2009 Copying metadata files into staging area for
FreeBSD/amd64 7.2-RELEASE
Mon Aug 24 12:16:59 PDT 2009 Constructing metadata index and tag for FreeBSD/amd64
7.2-RELEASE

Files found which include build stamps:
kernel|generic|/GENERIC/hptrr.ko
kernel|generic|/GENERIC/kernel
world|base|/boot/loader
world|base|/boot/pxeboot
world|base|/etc/mail/freebsd.cf
world|base|/etc/mail/freebsd.submit.cf
world|base|/etc/mail/sendmail.cf
world|base|/etc/mail/submit.cf
world|base|/lib/libcrypto.so.5
world|base|/usr/bin/ntpq
world|base|/usr/include/osreldate.h
world|base|/usr/lib/libalias.a
world|base|/usr/lib/libalias_cuseeme.a
world|base|/usr/lib/libalias_dummy.a
world|base|/usr/lib/libalias_ftp.a
...
```

Finally, the build completes.

```
Values of build stamps, excluding library archive headers:
v1.2 (Aug 25 2009 00:40:36)
v1.2 (Aug 25 2009 00:38:22)
@()FreeBSD 7.2-RELEASE 0: Tue Aug 25 00:38:29 UTC 2009
FreeBSD 7.2-RELEASE 0: Tue Aug 25 00:38:29 UTC 2009
    root@server.myhost.com:/usr/obj/usr/src/sys/GENERIC
7.2-RELEASE
Mon Aug 24 23:55:25 UTC 2009
Mon Aug 24 23:55:25 UTC 2009
 built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
 built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
 built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
 built by root@server.myhost.com on Tue Aug 25 00:16:15 UTC 2009
Mon Aug 24 23:46:47 UTC 2009
ntpq 4.2.4p5-a Mon Aug 24 23:55:53 UTC 2009 (1)
 * Copyright (c) 1992-2009 The FreeBSD Project.
Mon Aug 24 23:46:47 UTC 2009
Mon Aug 24 23:55:40 UTC 2009
Aug 25 2009
ntpd 4.2.4p5-a Mon Aug 24 23:55:52 UTC 2009 (1)
ntpdate 4.2.4p5-a Mon Aug 24 23:55:53 UTC 2009 (1)
ntpdc 4.2.4p5-a Mon Aug 24 23:55:53 UTC 2009 (1)
Tue Aug 25 00:21:21 UTC 2009
Tue Aug 25 00:21:21 UTC 2009
Tue Aug 25 00:21:21 UTC 2009
Mon Aug 24 23:46:47 UTC 2009

FreeBSD/amd64 7.2-RELEASE initialization build complete.  Please
review the list of build stamps printed above to confirm that
they look sensible, then run
 sh -e approve.sh amd64 7.2-RELEASE
to sign the release.
```

Approve the build if everything is correct. More information on determining this can be found in the distributed source file named USAGE. Execute scripts/approve.sh, as directed. This will sign the release, and move components into a staging area suitable for uploading.

```
# cd /usr/local/freebsd-update-server
# sh scripts/mountkey.sh
```

```
# sh -e scripts/approve.sh amd64 7.2-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Signing build for FreeBSD/amd64 7.2-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to patch source directories for
FreeBSD/amd64 7.2-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to upload staging area for FreeBSD/amd64
7.2-RELEASE
Wed Aug 26 12:50:07 PDT 2009 Updating databases for FreeBSD/amd64 7.2-RELEASE
```

```
Wed Aug 26 12:50:07 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.2-RELEASE
```

After the approval process is complete, the upload procedure may be started.

```
# cd /usr/local/freebsd-update-server
# sh scripts/upload.sh amd64 7.2-RELEASE
```

In the event update code needs to be re-uploaded, this may be done by changing to the public distributions directory for the target release and updating attributes of the *uploaded* file.

```
# cd /usr/local/freebsd-update-server/pub/7.2-RELEASE/amd64
# touch -t 200801010101.01 uploaded
```

The uploaded files will need to be in the document root of the webserver in order for updates to be distributed. The exact configuration will vary depending on the web server used. For the Apache web server, please refer to the Configuration of Apache servers section in the Handbook.

Update client's `KeyPrint` and `ServerName` in /etc/freebsd-update.conf, and perform updates as instructed in the FreeBSD Update section of the Handbook.

In order for FreeBSD Update Server to work properly, updates for both the *current* release and the release *one wants to upgrade to* need to be built. This is necessary for determining the differences of files between releases. For example, when upgrading a FreeBSD system from 7.1-RELEASE to 7.2-RELEASE, updates will need to be built and uploaded to your distribution server for both versions.

For reference, the entire run of init.sh is attached.

6. Building a Patch

Every time a security advisory or security notice is announced, a patch update can be built.

For this example, 7.1-RELEASE will be used.

A couple of assumptions are made for a different release build:

- Setup the correct directory structure for the initial build.
- Perform an initial build for 7.1-RELEASE.

Create the patch directory of the respective release under /usr/local/freebsd-update-server/patches/.

```
% mkdir -p /usr/local/freebsd-update-server/patches/7.1-RELEASE/
% cd /usr/local/freebsd-update-server/patches/7.1-RELEASE
```

As an example, take the patch for named(8). Read the advisory, and grab the necessary file from FreeBSD Security Advisories. More information on interpreting the advisory, can be found in the FreeBSD Handbook.

In the security brief, this advisory is called SA-09:12.bind. After downloading the file, it is required to rename the file to an appropriate patch level. It is suggested to keep this consistent with official FreeBSD patch levels, but its name may be freely chosen. For this build, let us follow the currently established practice of FreeBSD and call this p7. Rename the file:

```
% cd /usr/local/freebsd-update-server/patches/7.1-RELEASE/; mv bind.patch 7-SA-
09:12.bind
```

When running a patch level build, it is assumed that previous patches are in place. When a patch build is run, it will run all patches contained in the patch directory.

There can be custom patches added to any build. Use the number zero, or any other number.

It is up to the administrator of the FreeBSD Update Server to take appropriate measures to verify the authenticity of every patch.

At this point, a *diff* is ready to be built. The software checks first to see if a scripts/init.sh has been run on the respective release prior to running the diff build.

```
# cd /usr/local/freebsd-update-server
# sh scripts/diff.sh amd64 7.1-RELEASE 7
```

What follows is a sample of a *differential* build run.

```
# sh -e scripts/diff.sh amd64 7.1-RELEASE 7
Wed Aug 26 10:09:59 PDT 2009 Extracting world+src for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 17:10:25 UTC 2009 Building world for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 18:05:11 UTC 2009 Distributing world for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 18:06:16 UTC 2009 Building and distributing kernels for FreeBSD/amd64 7.1-
RELEASE-p7
Wed Aug 26 18:17:50 UTC 2009 Constructing world components for FreeBSD/amd64 7.1-
RELEASE-p7
Wed Aug 26 18:18:02 UTC 2009 Distributing source for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 11:19:23 PDT 2009 Moving components into staging area for FreeBSD/amd64
7.1-RELEASE-p7
Wed Aug 26 11:19:37 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 11:19:42 PDT 2009 Indexing world0 for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 11:23:02 PDT 2009 Extracting world+src for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 18:23:29 UTC 2010 Building world for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 19:18:15 UTC 2010 Distributing world for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 19:19:18 UTC 2010 Building and distributing kernels for FreeBSD/amd64 7.1-
RELEASE-p7
```

```
Thu Sep 30 19:30:52 UTC 2010 Constructing world components for FreeBSD/amd64 7.1-
RELEASE-p7
Thu Sep 30 19:31:03 UTC 2010 Distributing source for FreeBSD/amd64 7.1-RELEASE-p7
Thu Sep 30 12:32:25 PDT 2010 Moving components into staging area for FreeBSD/amd64
7.1-RELEASE-p7
Wed Aug 26 12:32:39 PDT 2009 Extracting extra docs for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:32:43 PDT 2009 Indexing world1 for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:35:54 PDT 2009 Locating build stamps for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:36:58 PDT 2009 Reverting changes due to build stamps for FreeBSD/amd64
7.1-RELEASE-p7
Wed Aug 26 12:37:14 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:37:14 PDT 2009 Preparing to copy files into staging area for
FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:37:15 PDT 2009 Copying data files into staging area for FreeBSD/amd64
7.1-RELEASE-p7
Wed Aug 26 12:43:23 PDT 2009 Copying metadata files into staging area for
FreeBSD/amd64 7.1-RELEASE-p7
Wed Aug 26 12:43:25 PDT 2009 Constructing metadata index and tag for FreeBSD/amd64
7.1-RELEASE-p7
...
Files found which include build stamps:
kernel|generic|/GENERIC/hptrr.ko
kernel|generic|/GENERIC/kernel
world|base|/boot/loader
world|base|/boot/pxeboot
world|base|/etc/mail/freebsd.cf
world|base|/etc/mail/freebsd.submit.cf
world|base|/etc/mail/sendmail.cf
world|base|/etc/mail/submit.cf
world|base|/lib/libcrypto.so.5
world|base|/usr/bin/ntpq
world|base|/usr/include/osreldate.h
world|base|/usr/lib/libalias.a
world|base|/usr/lib/libalias_cuseeme.a
world|base|/usr/lib/libalias_dummy.a
world|base|/usr/lib/libalias_ftp.a
...
Values of build stamps, excluding library archive headers:
v1.2 (Aug 26 2009 18:13:46)
v1.2 (Aug 26 2009 18:11:44)
@()FreeBSD 7.1-RELEASE-p7 0: Wed Aug 26 18:11:50 UTC 2009
FreeBSD 7.1-RELEASE-p7 0: Wed Aug 26 18:11:50 UTC 2009
    root@server.myhost.com:/usr/obj/usr/src/sys/GENERIC
7.1-RELEASE-p7
Wed Aug 26 17:29:15 UTC 2009
Wed Aug 26 17:29:15 UTC 2009
 built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
 built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
 built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
 built by root@server.myhost.com on Wed Aug 26 17:49:58 UTC 2009
Wed Aug 26 17:20:39 UTC 2009
```

```
ntpq 4.2.4p5-a Wed Aug 26 17:29:42 UTC 2009 (1)
 * Copyright (c) 1992-2009 The FreeBSD Project.
Wed Aug 26 17:20:39 UTC 2009
Wed Aug 26 17:29:30 UTC 2009
Aug 26 2009
ntpd 4.2.4p5-a Wed Aug 26 17:29:41 UTC 2009 (1)
ntpdate 4.2.4p5-a Wed Aug 26 17:29:42 UTC 2009 (1)
ntpdc 4.2.4p5-a Wed Aug 26 17:29:42 UTC 2009 (1)
Wed Aug 26 17:55:02 UTC 2009
Wed Aug 26 17:55:02 UTC 2009
Wed Aug 26 17:55:02 UTC 2009
Wed Aug 26 17:20:39 UTC 2009
...
```

Updates are printed, and approval is requested.

```
New updates:
kernel|generic|/GENERIC/kernel.symbols|f|0|0|0555|0|7c8dc176763f96ced0a57fc04e7c1b8d79
3f27e006dd13e0b499e1474ac47e10|
kernel|generic|/GENERIC/kernel|f|0|0|0555|0|33197e8cf15bbbac263d17f39c153c9d489348c2c5
34f7ca1120a1183dec67b1|
kernel|generic|/|d|0|0|0755|0||
src|base|/|d|0|0|0755|0||
src|bin|/|d|0|0|0755|0||
src|cddl|/|d|0|0|0755|0||
src|contrib|/contrib/bind9/bin/named/update.c|f|0|10000|0644|0|4d434abf0983df9bc474356
70d307fa882ef4b348ed8ca90928d250f42ea0757|
src|contrib|/contrib/bind9/lib/dns/openssldsa_link.c|f|0|10000|0644|0|c6805c39f3da2a06
dd3f163f26c314a4692d4cd9a2d929c0acc88d736324f550|
src|contrib|/contrib/bind9/lib/dns/opensslrsa_link.c|f|0|10000|0644|0|fa0f7417ee9da42c
c8d0fd96ad24e7a34125e05b5ae075bd6e3238f1c022a712|
...
FreeBSD/amd64 7.1-RELEASE update build complete.  Please review
the list of build stamps printed above and the list of updated
files to confirm that they look sensible, then run
 sh -e approve.sh amd64 7.1-RELEASE
to sign the build.
```

Follow the same process as noted before for approving a build:

```
# sh -e scripts/approve.sh amd64 7.1-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Signing build for FreeBSD/amd64 7.1-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to patch source directories for
FreeBSD/amd64 7.1-RELEASE
Wed Aug 26 12:50:06 PDT 2009 Copying files to upload staging area for FreeBSD/amd64
7.1-RELEASE
Wed Aug 26 12:50:07 PDT 2009 Updating databases for FreeBSD/amd64 7.1-RELEASE
Wed Aug 26 12:50:07 PDT 2009 Cleaning staging area for FreeBSD/amd64 7.1-RELEASE
```

```
The FreeBSD/amd64 7.1-RELEASE update build has been signed and is
ready to be uploaded.  Remember to run
 sh -e umountkey.sh
to unmount the decrypted key once you have finished signing all
the new builds.
```

After approving the build, upload the software:

```
# cd /usr/local/freebsd-update-server
# sh scripts/upload.sh amd64 7.1-RELEASE
```

For reference, the entire run of diff.sh is attached.

7. Tips

- If a custom release is built using the native `make release` procedure, freebsd-update-server code will work from your release. As an example, a release without ports or documentation can be built by clearing functionality pertaining to documentation subroutines `findextradocs ()`, `addextradocs ()` and altering the download location in `fetchiso ()`, respectively, in scripts/build.subr. As a last step, change the sha256(1) hash in build.conf under your respective release and architecture and you are ready to build off your custom release.

  ```
  # Compare ${WORKDIR}/release and ${WORKDIR}/$1, identify which parts
  # of the world|doc subcomponent are missing from the latter, and
  # build a tarball out of them.
  findextradocs () {
  }
  # Add extra docs to ${WORKDIR}/$1
  addextradocs () {
  }
  ```

- Adding `-j NUMBER` flags to `buildworld` and `obj` targets in the scripts/build.subr script may speed up processing depending on the hardware used, however it is not necessary. Using these flags in other targets is not recommended, as it may cause the build to become unreliable.

  ```
             # Build the world
           log "Building world"
           cd /usr/src &&
           make -j 2 ${COMPATFLAGS} buildworld 2>&1
        # Distribute the world
           log "Distributing world"
           cd /usr/src/release &&
           make -j 2 obj &&
           make ${COMPATFLAGS} release.1 release.2 2>&1
  ```

- Create an appropriate DNS SRV record for the update server, and put others behind it with variable weights. Using this facility will provide update mirrors, however this tip is not necessary unless you wish to provide a redundant service.

```
_http._tcp.update.myserver.com.     IN SRV   0 2 80   host1.myserver.com.
                      IN SRV   0 1 80   host2.myserver.com.
                      IN SRV   0 0 80   host3.myserver.com.
```