

# Update: Automating Ports Tree Updates

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME
	2015-11-04		WB

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Update</b>	<b>1</b>
<b>3</b>	<b>The Script: <i>update</i></b>	<b>1</b>

© 2015 Warren Block

Last updated 2015-11-04

Available in [HTML](#) or [PDF](#). Links to all my articles [here](#). Created with [AsciiDoc](#).

Updating the ports tree on FreeBSD.

## 1 Introduction

There are several steps to updating the ports tree and checking for installed applications that need to be updated. This can be simplified by automating it.

## 2 Update

This script is meant to self-configure. It expects the ports tree to be in */usr/ports*, although that can be overridden in the script.

The ports tree is checked to determine whether Subversion or *portsnap(8)* should be used to update it. I almost always use Subversion, so the *portsnap(8)* has not really been tested.

If ezjails are present, their ports tree is also updated.

The date of last update is shown for both the host and ezjails to make it easy to check */usr/ports/UPDATING*.

The script must be run as root. I call it *update*.

## 3 The Script: *update*

```
#!/bin/sh

# Warren Block
# last update 2015-11-04

PORTSDIR="/usr/ports"

AWK="/usr/bin/awk"
BASENAME="/usr/bin/basename"
DATE="/bin/date"
EZJAILADMIN="/usr/local/bin/ezjail-admin"
EZJAILCFG="/usr/local/etc/ezjail"
GREP="/usr/bin/grep"
JAILDIR="/usr/jails"
JAILLS=""
JLS="/usr/sbin/jls"
LS="/bin/ls"
PKG="/usr/local/sbin/pkg"
PORTMASTER="/usr/local/sbin/portmaster"
PORTSNAP="/usr/sbin/portsnap"
PRINTF="/usr/bin/printf"
SORT="/usr/bin/sort"
SVN="/usr/local/bin/svn"
SVNLITE="/usr/bin/svnlite"
TAIL="/usr/bin/tail"

DATEFMT="+%Y-%m-%d"
LOG="/tmp/update.log"
PKGDBDIR="/var/db/pkg"
```

```

UPDATEMETHOD=""

startlog() {
    echo "update log ${LOG} started `${DATE} ${DATEFMT}`" >> "${LOG}"
}

showit() {
    echo "$@" >> "${LOG}"
    echo "$@"
}

portsnaporsvn() {
    UPDATEMETHOD="svn"
    ${SVNLITE} info "${PORTSDIR}" 2>&1 1>/dev/null
    if [ $? -ne 0 ]; then
        UPDATEMETHOD="portsnap"
    fi
}

checkrequirements() {
    if [ "${UPDATEMETHOD}" = "svn" ]; then
        if [ ! -f "${SVN}" ]; then
            echo " ** ${SVN} not found, cannot update host"
            exit 1
        fi
    elif [ "${UPDATEMETHOD}" = "portsnap" ]; then
        if [ ! -f "${PORTSNAP}" ]; then
            echo " ** ${PORTSNAP} not found, cannot update host"
            exit 1
        fi
    fi

    if [ ! -f "${PORTMASTER}" ]; then
        echo " ** ${PORTMASTER} not found, cannot check host"
        exit 1
    fi

    if [ ! -f "${PKG}" ]; then
        echo " ** ${PKG} not found, cannot check package database"
        exit 1
    fi
}

updatehost() {
    showit "updating host with ${UPDATEMETHOD}"
    case ${UPDATEMETHOD} in
        svn)      ${SVN} up "${PORTSDIR}" 2>&1 >>"${LOG}" ;;
        portsnap) ${PORTSNAP} fetch update -p "${PORTSDIR}/" 2>&1 >> "${LOG}" ;;
    esac
    if [ $? -ne 0 ]; then
        showit " ** error on host update. log file: ${LOG}"
        exit 1
    fi
}

checkhost() {
    # use --index-only for faster test on the host, although it may not
    # be up to date with the actual ports themselves
    showit "checking host"

    OUTPUT=`${PORTMASTER} -L --index-only | egrep '(ew|ort) version|total install'`
    RC=$?
}

```

```
echo "${OUTPUT}" >> "${LOG}"
if [ "${RC}" -ne 0 ]; then
    showit " ** error checking host. log file: ${LOG}"
    exit 1
fi

if echo "${OUTPUT}" | ${GREP} -q "no new versions available" ; then
    # no updates needed
else
    OUTPUT=`echo "${OUTPUT}" | ${AWK} '{ print "\t", $0 }'`
    showit "${OUTPUT}"
    LASTUPDATE=$( ${PRINTF} "\tLast update: " ; ${DATE} -r `${PKG} query %t | ${SORT} | ${ ←
        TAIL} -nl`${DATEFMT} )
    showit "${LASTUPDATE}"
fi
}

locatejails() {
    if [ ! -d "${JAILEDIR}" ]; then
        echo "no jailedir ${JAILEDIR}, skipping jail check" >> "${LOG}"
        return
    fi

    if [ ! -d "${EZJAILCFGs}" ]; then
        echo "no ezjail config dir ${EZJAILCFGs}, skipping jail check" >> "${LOG}"
        return
    fi

    cfgs=`${LS} ${EZJAILCFGs}/* 2>/dev/null`

    for jail in ${cfgs}; do
        jail=`${BASENAME} "${jail}"`
        JAILS="${JAILS} ${jail%.norun}"
    done

    if [ -n "${JAILS}" ]; then
        return 0
    else
        echo "no jails found" >> "${LOG}"
        return 1
    fi
}

updatebasejail() {
    if [ -z "${JAILS}" ]; then
        return
    fi

    if [ -f "${EZJAILADMIN}" ]; then
        # update ports in the basejail
        showit "updating basejail"
        ${EZJAILADMIN} update -P >> "${LOG}"
    else
        showit " ** ${EZJAILADMIN} not found, cannot update basejail"
        return 1
    fi
}

checkjails() {
    for jail in ${JAILS}; do
        echo -n "jail: ${jail}"
    done
}
```

```
echo -n "jail: ${jail}" >> "${LOG}"
if ${JLS} name | ${GREP} -q -x ${jail} ; then
    showit
else
    showit ' (not running) '
fi

PKGDB="${JAILEDIR}/${jail}${PKGDBDIR}/local.sqlite"
if [ ! -f "${PKGDB}" ]; then
    showit " ** no package database at ${PKGDB}, cannot check port status"
    continue
fi

export PKG_DBDIR="${JAILEDIR}/${jail}${PKGDBDIR}"
OUTPUT='${PORTMASTER} -L | egrep '(ew|ort) version|total install''

if echo "${OUTPUT}" | ${GREP} -q "no new versions available" ; then
    # no updates needed
else
    OUTPUT='echo "${OUTPUT}" | ${AWK} '{ print "\t", $0 }''
    showit "${OUTPUT}"
    LASTUPDATE=$( ${PRINTF} "\tLast update: " ; ${DATE} -r '${PKG} query %t | ${SORT} | $ ←
        {TAIL} -n1 ` ${DATEFMT} )
    showit "${LASTUPDATE}"
fi
done
}

startlog
portsnaporsvn
checkrequirements
updatehost
checkhost
if locatejails; then
    updatebasejail
    checkjails
fi
```