

```
#!/bin/bash
#Usage: imgrm <path>
```

```
if [[ "$1" == "" ]] ; then Path="/" ; fi
```

```
_LCODES='Afrikaans|af%Albanian|alb|sq|sq0|sqi%Amharic|amh|am0|am%Arabic|ara|ar0|ar%Armenian|arm|hy|hy0|hye%
Australian|0au%Austrian|0at%Basque|eu%Belgian|0ch%Bosnian|bs%British|0gb%Bulgarian|bul|bg0|bg%Byelorussian|be|bel%
Canadian|0ca%Catalan|ca|ca0|cat%Cherokee|chr%Chinese|chi|zh|zh0|zho%Croatian|hrv|hr0|hr%Czech|cze|cs|ces|cs0|cz%
Danish|dan|da0|da%Dutch|dut|du|nld|nl0|nl%Emoji%English|en|eng|en0US%EN~Hybrid|0ca|0ie|0jp|0in|0nz|0sg|0uk|0za%
Estonian|est|et0|et%Finnish|fin|fi0|fi|se%Filipino|ph|fil|tl%French|fra|fr|fre|0fr%Gaelic|ga0|ga%German|deu|de|ger|0de|gsw%
Greek|gre|el|el0|el%Guarani|gl%Gujarati|guj|gu0|gu%Hawaiian|haw%Hebrew|heb|he|he0|iw%Hindi|hin|hi0|hi|bn|kn|ml|mr|te
%Hungarian|hun|hu0|hu%Icelandic|ice|is|is0|isl%Indonesian|id|ind|id0|in%Iranian|ira|fa|fa0|per%Italian|ita|it0|it%Japanese|ja|
ja0|jpn|ka|Kana%Kanuri|kr%Korean|kor|ko0|ko%Latvian|lav|lv0|lv%Lingala|ln%Lithuanian|lit|lt0|lt%Macedonian|mk|mk0|mac%
Malay|Malaysian|may|ms0|ms%Norwegian|nb|nor|no|nob|nn|nn0|nno%Panjabi|pa|pa0|pan%Polish|pol|pl0|pl%Portuguese|
por|pt0|pt%Romanian|ro|rum|mo|ro0|ron%Russian|ru|ru0|rus%Serbian|sr0|sr%Slovak|slk|sk|sk0|slo%Slovenian|slv|sl0|sl%
Spanish|spa|sp|sp0|es|0es|gl%SP~Hybrid|0419|0ar|0bo|0cl|0co|0cr|0do|0ec|0gt|0hn|0mx|0ni|0pa|0pe|0pr|0py|0sv|0us|
0uy|0ve%Swahili|sw%Swedish|swe|sv0|sv%Tamil|ta%Thai|tha|th0|th%Tibetan|bo%Turkish|tur|tr0|tr%Ukrainian|ukr|uk0|uk%
Urdu|ur%Vietnamese|vi|vt|vi0|vn0|vn%Welsh|cy%Zulu|zu'
```

```
#-----
_IDAEMONS='accessoryd%iPhone docks, AV cables & accessories|AddressBook%Cache Contacts for faster
loading|AdminLite%Force-Quit unresponsive processes|AOSNotification%MobileMe & iCloud over-the-air
sync|appsupport.cplogd%Logs 3rdParty app crashes|apsd%Push notifications|apsd.tcpdump.en0%Logs push notification
errors|apsd.tcpdump.pdp_ip0%Also logs push errors|aslmanager% System Log manager%|assitivetouchd%AssistiveTouch.app
helper|awd_iceX%2G & 3G use ice2; 4G+ uses ice3|certui.relay%Unverified website warnings|chud.chum% CHUD
tools|chud.pilotfish%CHUD for developers|crash_mover%Move Crashlogs to Mob/Lib/Logs|CrashHousekeeping%Deals with
crashes|daily%AppStore automatic update check|dataaccess.dataaccessd%MobileMe,Exchange,LDAP
Sync|datamigrator%Transfer SIM contacts to phone|DumpBasebandCrash%Dumps baseband crashes|DumpPanic%Writes
panic-info to Crash Logs|gamed%Needed only for GameCenter|graphicservices.sample%Display album
artwork|iapd%Manages accessories & Nike+iPod|iagent%Unknown but removal ok|itunescloudd%iTunes Cloud & Home
Sharing|livetrace%MobileMe tracking Daemon|marcoagent%Logs iMessage & FaceTime crashes|marco.tcpdump.en0%Logs
errors with WiFi connections|mobile.accessory_device_arbitrator%Accessories|mobile.obliteration%Settings Reset,
Data-wipe|mobile.profile_janitor%Provisioning profiles|mobile.softwareupdated%|jailbreak made this
useless|MobileInternetSharing%Internet Tethering|nmbd%Samba daemon?|notifyd%Needed only for Push
Notification|OTACrashCopier%Moves OTA crashlogs|OTATaskingAgent%OTA update periodic checks|powerlog%Logs
incompatible chargers|printd%Starts the AirPrint daemon|profilepolicy%For iOS5; Unknown but removal OK|psctl%Camera
Connection Kit support|racon%VPN Virtual Private Networks|ReportCrash.DirectoryService% Crash report
1|ReportCrash.Jetsam% Crash report 2|ReportCrash% Crash report 3|ReportCrash.SafetyNet% Crash report
4|ReportCrash.SimulateCrash% Crash report 5|ReportCrash.StackShot% Crash report 6|scrod%Voice
Control|search.appindexer%Indexes apps for Spotlight|searchd%Remove to disable Spotlight search|sharktrace%Developer
crash logs|SpringBoard.SetupHelper%iOS Setup app first-run|stackshot.server%Unknown but removal
ok|storage_mounter%iPad Camera Connection Kit|syslogd%Logs system events|tcpdump.server%Dump traffic on a
network|usbethernetsharing%|vibrationmanagerd%Custom vibration patterns|voiced%Voice Control|VoiceOverTouch%Also
Voice Control|vsassetd%Starts Voice control|wapic%Chinese WiFi error log|com.saurik.Cydia.Startup%Cydia AutoInstall of DEB
files'
```

```
#-----
_SIMS="ae|U.A.Emirates al|Albania am|Armenia ar|Argentina at|Austria au|Australia ae|Belgium bg|Bulgaria bh|Bahrain
bo|Bolivia br|Brazil bw|Botswana ca|Canada cf|C.A.Rep ch|Switzerland ci|Cote_d'Ivoire cl|Chile cm|Cameroon cn|China
co|Colombia cr|Costa_Rica cz|Czech_Republic de|Germany Germany|Germany dk|Denmark do|Dominican_Rep ec|Ecuador
ee|Estonia eg|Egypt es|Spain fi|Finland fr|France France|France gn|Guinea gr|Greece gt|Guatemala gu|Guam hk|Hong_Kong
hn|Honduras hr|Croatia hu|Hungary id|Indonesia ie|Ireland il|Israel in|India it|Italy Italy|Italy jm|Jamaica jo|Jordan jp|Japan
ke|Kenya kr|South_Korea kw|Kuwait li|Liechtenstein lt|Lithuania lu|Luxembourg lv|Latvia ma|Morocco md|Moldova
me|Montenegro mg|Madagascar mk|Macedonia ml|Mali mo|Macau mq|Martinique mt|Malta mu|Mauritius mx|Mexico
my|Malaysia ne|Niger ni|Nicaragua nl|Netherlands no|Norway nz|New_Zealand pa|Panama pe|Peru ph|Philippines pl|Poland
pt|Portugal py|Paraguay qa|Qatar re|Reunion ro|Romania ru|Russia sa|Saudi_Arabia se|Sweden sg|Singapore si|Slovenia
sk|Slovakia sn|Senegal sv|El_Salvador th|Thailand tn|Tunisia tr|Turkey tt|Trinidad_&_Tobago tw|Taiwan ug|Uganda
uk|England uy|Uruguay ve|Venezuela vn|Vietnam wi|Caribbean za|South_Africa"
```

```
#-----
_RCODES="AF|Afghanistan%AL|Albania%DZ|Algeria%AS|American_Samoa%AD|Andorra%AO|Angola%AI|Anguilla%AQ|
Antarctica%AG|Antigua_&_Barbuda%AR|Argentina%AM|Armenia%AW|Aruba%AU|Australia%AT|Austria%AZ|Azerbaijan%BS|
Bahamas%BH|Bahrain%BD|Bangladesh%BB|Barbados%BY|Belarus%BE|Belgium%BZ|Belize%BJ|Benin%BM|Bermuda%BT|Bhutan%BO
|Bolivia%BA|Bosnia_&_Herzegovina%BW|Botswana%BR|Brazil%BN|Brunei%BG|Bulgaria%BF|Burkina_Faso%BI|Burundi%KH|
Cambodia%CM|Cameroon%CA|Canada%CV|Cape_Verde%KY|Cayman_Islands%CF|Central_African_Republic%TD|Chad%CL|Chile
%CN|China%CO|Colombia%KM|Comoros%CG|Congo-Brazzaville%CD|DR_Congo%CR|Costa_Rica%CI|Côte_D'Ivoire%HR|Croatia
%CU|Cuba%CW|Curaçao%CY|Cyprus%CZ|Czech_Republic%DK|Denmark%DJ|Djibouti%DM|Dominica%DO|Dominican_Republic%
EC|Ecuador%EG|Egypt%SV|El_Salvador%GQ|Equatorial_Guinea%ER|Eritrea%EE|Estonia%ET|Ethiopia%FK|Falkland_Islands%FO|
Faroe_Islands%FJ|Fiji%FI|Finland%FR|France%GF|French_Guiana%PF|French_Polynesia%GA|Gabon%GM|Gambia%GE|Georgia%DE
|Germany%GH|Ghana%GI|Gibraltar%GR|Greece%GL|Greenland%GD|Grenada%GP|Guadeloupe%GU|Guam%GT|Guatemala%GG|
Guernsey%GN|Guinea%GW|Guinea-Bissau%GY|Guyana%HT|Haiti%VA|Vatican_City%HN|Honduras%HK|Hong_Kong%HU|
Hungary%IS|Iceland%IN|India%ID|Indonesia%IR|Iran%IQ|Iraq%IE|Ireland%IM|Isle_Of_Man%IL|Israel%IT|Italy%JM|Jamaica%JP|Japan
```

```
%E|Jersey%JO|Jordan%KZ|Kazakhstan%KE|Kenya%KP|North_Korea%KR|South_Korea%KW|Kuwait%KG|Kyrgyzstan%LA|Lao%LV|
Latvia%LB|Lebanon%LS|Lesotho%LR|Liberia%LY|Libya%LI|Liechtenstein%LT|Lithuania%LU|Luxembourg%MO|Macao%MK|
Macedonia%MG|Madagascar%MW|Malawi%MY|Malaysia%MV|Maldives%ML|Mali%MT|Malta%MH|Marshall_Islands%MQ|
Martinique%MR|Mauritania%MU|Mauritius%MX|Mexico%FM|Micronesia%MD|Moldova%MC|Monaco%MN|Mongolia%ME|
Montenegro%MS|Montserrat%MA|Morocco%MZ|Mozambique%MM|Myanmar%NA|Namibia%NP|Nepal%NL|Netherlands%NC|
New_Caledonia%NZ|New_Zealand%NI|Nicaragua%NE|Niger%NG|Nigeria%NO|Norway%OM|Oman%PK|Pakistan%PS|
Palestinian_Territory%PA|Panama%PG|Papua_New_Guinea%PY|Paraguay%PE|Peru%PH|Philippines%PN|Pitcairn%PL|Poland%PT|
Portugal%PR|Puerto_Rico%QA|Qatar%RE|Réunion%RO|Romania%RU|Russia%RW|Rwanda%BL|Saint_Barthélemy%SH|Saint_Helena
%KN|Saint_Kitts%LC|Saint_Lucia%MF|French_Saint_Martin%PM|Saint_Pierre_&_Miquelon%VC|Saint_Vincent%WS|Samoa%SM|
San_Marino%ST|Sao_Tome_&_Principe%SA|Saudi_Arabia%SN|Senegal%RS|Serbia%SC|Seychelles%SL|Sierra_Leone%SG|Singapore
%SX|Dutch_Saint_Maarten%SK|Slovakia%SI|Slovenia%SB|Solomon_Islands%SO|Somalia%ZA|South_Africa%SS|South_Sudan%ES|
Spain%LK|Sri_Lanka%SD|Sudan%SR|Suriname%SZ|Swaziland%SE|Sweden%CH|Switzerland%SY|Syria%TW|Taiwan%TJ|Tajikistan%
TZ|Tanzania%TH|Thailand%TL|East_Timor%TG|Togo%TT|Trinidad_&_Tobago%TN|Tunisia%TR|Turkey%TM|Turkmenistan%TC|
Turks_&_Caicos%UG|Uganda%UA|Ukraine%AE|United_Arab_Emirates%GB|United_Kingdom%US|United_States%UY|Uruguay%UZ
|Uzbekistan%VE|Venezuela%VN|Vietnam%VG|British_Virgin_Islands%VI|U.S_Virgin_Islands%WF|Wallis_&_Futuna%EH|
Western_Sahara%YE|Yemen%ZM|Zambia%ZW|Zimbabwe"
```

```
#-----
_RD='\e[0;31m';_GR='\e[0;32m';_YL='\e[0;33m';_BL='\e[0;34m';_MG='\e[0;35m';_CY='\e[0;36m';_WH='\e[0;37m';
_RS='\e[0;0m';function red() { printf "${_RD}"; };function green() { printf "${_GR}"; };function yellow() { printf "${_YL}"; };function
blue() { printf "${_BL}"; };function magenta() { printf "${_MG}"; };function cyan() { printf "${_CY}"; };function white() { printf
"${_WH}"; };function nocolor() { printf "${_RS}"; };mkdir -p /private/var/mobile/Media/Downloads/Logs;chown -R
mobile:mobile /private/var/mobile/Media/Downloads/Logs;function repeat() { n=$1;shift;while [ $(($n - 1)) -ge 0 ];do
"$@";done; };_WIDTH="$((tput cols))";_SYSLIB="/System/Library";_PRIVAR="/private/var";_MOBILE="/private/var/mobile";
_STAPPS="$(find /private/var/stash -name 'Applications*')";_MOBAPP="/private/var/mobile/Applications";
_LOGS="/private/var/mobile/Media/Downloads/Logs";_DOWNL="/private/var/mobile/Media/Downloads";_TF="/tmp/tf.*";
_4LINE="$(repeat ${_WIDTH}/4 printf '-');_DEL="$(printf '\b%.0s {1..12};printf '%.0s {1..12};printf '\b%.0s
{1..12}';function cleanup() { rm -f $_TF; };function blank() { printf "\r$(repeat $_WIDTH printf '\r')";};function bottom() {
printf "\n\033[1A";};function notab() { if [[ $_WIDTH -le 60 ]];then _NL='\n';_TB='\t';_LS='\r';_SP=' ';_BK='\b';
_AC="$_CY";_AY="$_YL";_AR="$_RD";_AB="$_BL";_AM="$_MG";"@|sed -e s:'^::g';else "$@";_ST='';fi;};function indent()
{ if [[ $_WIDTH -ge 60 ]];then "$@"|sed 's: ^::g';else "$@";fi;};function noindent() { if [[ $_WIDTH -le 60 ]];then "$@"|sed 's: ^::g';
else "$@";fi;};function allcaps() { if [[ $_WIDTH -ge 60 ]];then printf "$@";else printf "\n${@^^}";fi;};function cuttxt() {
if [[ $_WIDTH -ge 60 ]];then printf "$@";else nocolor;fi;};function hobble() { if [[ $_WIDTH -le 60 ]];then "$@" fn;else fn
;fi;};function pa2ste() { if [[ $_WIDTH -le 60 ]];then paste -;else paste - -;fi;};function pa3ste() { if [[ $_WIDTH -le 60 ]];then
paste - -;else paste - - -;fi;};function pa4ste() { if [[ $_WIDTH -le 60 ]];then paste - -;else paste - - - -;fi;};function
cutoff() { if [[ $_WIDTH -le 60 ]];then cut -c -60|sed 's/[ ]*[a-zA-Z0-9\./\%]$/g';else cut -c -120;fi;};function cutend() { if [[
$_WIDTH -le 60 ]];then cut -c -59;else cut -c -120;fi;};function shortd() { if [[ $_WIDTH -le 60 ]];then "$@"|sed -e 's: ^::g' -e 's: :$ (red):$ (nocolor):g' -e 's: %: :g';else "$@"|sed -e 's: ^::g' -e 's: :$ (red):$ (nocolor):g' -e 's: %: :g';
else "$@"|sed -e 's: :$ (red):$ (nocolor):g' -e 's: %: :g';fi;};function liner() { if [[ $_WIDTH -le 60 ]];then printf "${_WH}$(repeat 45 printf '-')\n${_WH}$(repeat 45 printf
'-)';else printf "\n ${_RS}---$_RD$(repeat 77 printf '-')${_RS}---\n";fi;};function bliner() { if [[ $_WIDTH -le 60 ]];then printf
"${_BL}$(repeat 45 printf '-')";else printf "\n ${_RS}---$_BL$(repeat 77 printf '-')${_RS}---\n";fi;};function gap() { if [[
$_WIDTH -le 60 ]];then printf "$_LS${_RD}---$_RS$(repeat 39 printf '-')${_RD}---";else printf "";fi;};function blugap() { if [[
$_WIDTH -le 60 ]];then printf "$_LS${_RS}---$_BL$(repeat 39 printf '-')${_RS}---";else printf "";fi;};function redgap() { if [[
$_WIDTH -le 60 ]];then printf "$_LS${_RS}---$_RD$(repeat 39 printf '-')${_RS}---";else printf "";fi;};function erata() { printf
"${_YL}INPUT${_RD}! \b${_WH}ERROR";sleep 1;printf "$_DEL";printf "${_YL}SELECTS${_WH}AGAIN";sleep 1;printf "$_DEL";
};function answer() { unset ii;while read -s -n 1 ii;do ii=${ii^^};if [[ "$ii" = Y || "$ii" = N ]];then break;fi;erata;done;};function
zanswer() { unset ff;while read -s -n 1 ff;do if [[ "$ff" = 1 || "$ff" = 0 ]];then break;fi;erata;done;};function yn() { notab
printf "${_RS}[$_GR]Y${_RS} or ${_RS}[$_RD]N${_RS}";etc;ask;};function oi() { notab printf "${_RS}[$_GR]O${_RS} or
${_RS}[$_RD]1${_RS}";etc;ask;};function print_yn() { case "$ii" in Y) _YN=" ${_RS}[$_GR]Y${_RS}";; N) _YN="
${_RS}[$_RD]N${_RS}";;esac;echo -e "$_YN";sleep 0.5;};function print_oi() { case "$ff" in 0) _OI=" ${_RS}[$_GR]O${_RS}";; 1)
_OI=" ${_RS}[$_RD]1${_RS}";;esac;echo -e "$_OI";sleep 0.5;};function notxt() { stty -echo;};function txt() { stty echo;
};function icount() { wc -l 2>/dev/null </tmp/tf.logrm.log;printf "$?"|tr -d '0'|tr '1' '0';};function icountmp() { wc -l
2>/dev/null </tmp/tf.count.log;printf "$?"|tr -d '0'|tr '1' '0';};function isize() { grep -o "[0-9\.*][GMKB]" /tmp/tf.size.log;
};function dot() { notab printf "$(repeat $(tput cols)/6 printf ' ') \.n" && sleep 0;};function etc() { sleep 0.1;cutxt '!';sleep
0.1;cutxt '-';sleep 0.1;cutxt '!';sleep 0.1;};function etcn() { sleep 0.1;cutxt '!';sleep 0.1;cutxt '!';sleep 0.1;cutxt '!';sleep
0.1;printf "\n";};function etcnn() { sleep 0.1;cutxt '!';sleep 0.1;cutxt '!';sleep 0.1;cutxt '!';sleep 0.1;printf "\n";cutxt "\n";
};function ask() { printf '?';};
#-----
```

```
function xrem() { printf "${E%*}"${_XREM}|tr '\n'|grep '!|sort -fu|tr '\n'|sed 's:|:|'; }
function xkeep() { printf "${E%*}"${_XKEEP}|tr '\n'|grep '!|sort -fu|tr '\n'|sed 's:|:|'; }
function appname() { "$@"|sed -e 's:(Game):\1 Center:g' -e 's:(iOS):\1 Diagnostics:g'; }
function xsub() { "$@"|sed -e 's:Game_*[a-zA-Z]*:Game:g' -e 's:Sports_[a-zA-Z]*:Sports:g'; }
function updatxcpt() { grep -Ev '_XREM|_XKEEP' ${_LOGS}/rm.logxcpt.log 2>/dev/null > /tmp/tf.xc1.log;echo
_XREM$(xrem)|grep -v '^_XREM|$'|sed 's:|:|' >> /tmp/tf.xc1.log;echo _XKEEP$(xkeep)|sed -e 's:(($echo
${_XREM}|sed 's:|:|g):|:|'|grep -v '^_XKEEP|$'|sed 's:|:|'| tr -s '| ' >> /tmp/tf.xc1.log;grep '!| /tmp/tf.xc1.log >
${_LOGS}/rm.logxcpt.log 2>/dev/null ; }
#-----
function editlist() { cat /tmp/tf.${_TLOG}.log 2>/dev/null > /tmp/tf.${_TLOG}_2.log ; notab echo -ne "\n ${_WH}Do you want
```

```
to ${_RD}EDIT${_WH} this list? ${_NL}${_RS}";etc;yn;answer;print_yn;if [[ "$ii" = N ]];then blank;printf "\033[1A\033[1A";elif [[
"$ii" = Y ]];then notab echo -ne "\n ${_RS}Enter ${_CY}number${_RS}(${_CY}${_BK}s${_RS})${_BK} ${_RS}separated by a
${_WH}single space ${_NL}${_RS}& hit [${_MG}ENTER${_RS}]${_YL}\n\n "; txt && read _XCEPT ;if [[ $_XCEPT = "" ]];then notab
printf "\033[1A\033[1A\033[1A";blank;notab echo -ne " Nothing selected so list was not edited\n";else for A in $_XCEPT ;do
A="$(echo $A|sed -e 's:\^0*:')";while [[ $A = *[^:digit:]* || ! $(sed -n "${A}p" /tmp/tf.${_TLOG}.log 2>/dev/null) ]];do notab
echo -ne "\n$(blank) ${_CY}${A} ${_RS}is not a valid selection number. ${_RD}RETYPE SELECTION${_RS}: ${_CY}";while read tt
;do tt="$(echo $tt|sed 's:\^0*:')";if [[ $(sed -n "${tt}p" /tmp/tf.${_TLOG}.log 2>/dev/null) ]];then break;fi;notab echo -ne
"\033[1A$(blank) ${_CY}${tt} ${_RS}is not a valid selection number. ${_RD}RETYPE SELECTION${_RS}: ${_CY}";done;
_XCEPT="$(echo $_XCEPT|tr ' '\n'|sort -fu|sed 's:${A}:${tt}:|tr '\n ' '); A="$tt";done;done; _XP="$(echo $_XCEPT|sed -e 's/
/p;/g)"; echo;if [[ $_XP ]];then notab echo -e "$(sed -n "${_XP}p" /tmp/tf.${_TLOG}.log|sed -e 's:*/,,' -e 's:\.([a-zA-Z])*\:'
-e 's:\^[0-9]*\:'|tr [a-z] [A-Z]|uniq|sed -e 's:%: :g' -e 's:\^: $(white)REMOVING $(nocolor)from LIST\ : $(yellow):' -e
's:~:$(magenta):)";sleep 1 ;_CC="$(sed -n "${_XP}p" /tmp/tf.${_TLOG}.2.log)"; _XD="$(echo $_XCEPT|sed -e 's/ /d;/g)";sed
"${_XD}d" /tmp/tf.${_TLOG}.log > /tmp/tf.${_TLOG}.tf.log ;mv /tmp/tf.${_TLOG}.tf.log /tmp/tf.${_TLOG}.log ;
_EPRINT="YES";fi;fi;notxt;unset _TLOG ;unset _XP ; }
```

```
function editname() { notab echo -ne "\n ${_WH}Do you want to ${_RD}EDIT${_WH} this list?
${_NL}${_RS}";etc;yn;answer;print_yn;if [[ "$ii" = N ]];then printf "\033[1A\033[1A";blank;elif [[ "$ii" = Y ]];then notab echo
-ne " ${_RS}Enter ${_CY}selection${_RS}(${_CY}${_BK}s${_RS})${_BK}${_RS} to ${_WH}exclude${_RS} from ${_CY}Removal
List${_NL} ${_RS}separated by a ${_WH}single space ${_RS}& hit [${_MG}ENTER${_RS}]${_YL}\n\n ";txt && read _XCEPT ;if [[
$_XCEPT = "" ]];then notab printf "\033[1A\033[1A\033[1A";blank;notab printf " Nothing selected so list was not
edited\n";else for A in $_XCEPT ;do while [ ! $(grep -Ei "[a-zA-Z0-9]*${A}[a-zA-Z.~]" /tmp/tf.${_TLOG}.log 2>/dev/null)
]];do notab echo -ne "\n\n\033[1A$(blank) ${_CY}${A} ${_RS}is not a valid selection. ${_RD}RETYPE SELECTION${_RS}:
${_CY}";while read tt ;do if grep -qi "$tt" /tmp/tf.${_TLOG}.log ;then break;fi;echo -ne "\033[1A$(blank) ${_CY}${tt} ${_RS}is
not a valid selection. ${_RD}RETYPE SELECTION${_RS}: ${_CY}";done; _XCEPT="$(echo $_XCEPT|tr ' '\n'|sort -fu|sed
's:${A}:${tt}:"); A="$tt";done;done; _X="$(echo $_XCEPT|tr ' '\n'|)";notab echo -e "\n${_YL}$(grep -Eio "$_X"
/tmp/tf.${_TLOG}.log|sed -e 's:*/,,' -e 's:\.([a-zA-Z])*\:'|tr [a-z] [A-Z]|uniq|sed -e 's:%: :g' -e 's:\^: $(white)REMOVING
$(nocolor)From LIST $(yellow):' -e 's:~:$(magenta):)";sleep 1 ;grep -Eiv "$(printf ${_X}|sed -e 's:(\<Nike\>)\ : \1|Victoria:|
-e 's:(\<AdSheet\>)\ : \1|Ad:I)' /tmp/tf.${_TLOG}.log > /tmp/tf.${_TLOG}.tf.log ;mv /tmp/tf.${_TLOG}.tf.log
/tmp/tf.${_TLOG}.log ; _EPRINT="YES";fi;fi;notxt;unset _TLOG ; }
```

```
function editprint() { if [[ "$_EPRINT" == "YES" ]];then if [ ! "$(echo $_PR1NT|grep [a-z])" ];then notab echo -ne "\n $(red)The
list is now empty$(nocolor) " ;etcn;else notab echo -e "\n ${_RD}Updated list${_RS}:\n" && echo -e "${_PR1NT}";fi;sleep 0.5
;unset _EPRINT ;unset _PR1NT ;else unset _EPRINT ;unset _PR1NT ;fi; }
```

```
function logxlang() { if [[ $_CC ]];then echo $_LCODES|sed -e 's:%:\n:g'|grep -Eiw "$(echo $_CC|tr ' '\n'|)";sed 's:%: :g' >
/tmp/tf.x1.log ;grep "_X${_XCPT}" $_LOGS}/rm.logxcept.log 2>/dev/null|sed -e 's:\.([A-Z])\ : \n\1:g'|grep -v "_X${_XCPT}"
>> /tmp/tf.x1.log;sort -fu /tmp/tf.x1.log|tr '\n ' '|';sed -e 's:\^:_X${_XCPT}:' -e 's:|:|:' > /tmp/tf.x2.log ;grep -v
"^_X${_XCPT}" $_LOGS}/rm.logxcept.log 2>/dev/null|sed 's:\^:\n:' >> /tmp/tf.x2.log ;grep '.' /tmp/tf.x2.log >
$_LOGS}/rm.logxcept.log && rm -f /tmp/tf.x*.log ;unset _CC ;fi; }
```

```
function logxlist() { if [[ $_CC ]];then echo "$_CC"|sed 's:~:*/:g' > /tmp/tf.x1.log ;grep "_X${_XCPT}" $_LOGS}/rm.logxcept.log
2>/dev/null|tr ' '\n'|sed 's:\^:_X${_XCPT}:'|grep '.' >> /tmp/tf.x1.log ;sort -fu /tmp/tf.x1.log|tr '\n ' '|';sed -e
's:\^:_X${_XCPT}:' -e 's:|:|:' > /tmp/tf.x2.log ;grep -v "^_X${_XCPT}" $_LOGS}/rm.logxcept.log 2>/dev/null|sed 's:\^:\n:' >>
/tmp/tf.x2.log ;grep '.' /tmp/tf.x2.log > $_LOGS}/rm.logxcept.log && rm -f /tmp/tf.x*.log ;unset _CC ;fi; }
```

```
_NAVOD="${_RS} ---${_RD}${_4LINE}${_WH}INSTRUCTIONS${_RD}${_4LINE}\b${_RS}---$(liner)${_YL}\n iDevice Theming, the
non-WinterBoard method\n ${_RS}Create Icon Source-folder '${_RD}iconography${_RS}' in
${_NL}${_CY}/private/var/mobile/Media/Downloads${_RS}\n Inside, create ${_WH}4 subfolders ${_RS}mirroring UI icon${_NL}
'relative' destinations\n Name subfolders to match the basename of ${_NL}destination directories${_RS} as
follows:$(bliner)$(gap)\n ${_RD}iconography${_RS}/${_WH}Stash_ ${_RS}${_AY}  ${_NL}mirrors
${_MG}/private/var/stash/Applications\t${_RS}${_NL}${_CY}root${_RS}:${_CY}admin${_RS}${_NL}$(redgap)\n
${_RD}iconography${_RS}/${_WH}Mobile_ ${_RS}${_AY}  ${_NL}mirrors
${_MG}/private/var/mobile/Applications\t${_RS}${_NL}${_CY}mob°${_RS}:${_CY}mobile${_RS}${_NL}$(redgap)\n
${_RD}iconography${_RS}/${_WH}System_ ${_RS}${_AY}  ${_NL}mirrors
${_MG}/System/Library\t\t\t${_RS}${_NL}${_CY}root${_RS}:${_CY}wheel${_RS}${_NL}$(redgap)\n
${_RD}iconography${_RS}/${_WH}Library_ ${_RS}${_AY}  ${_NL}mirrors
${_MG}/Library\t\t\t\t\t${_RS}${_NL}${_CY}root${_RS}:${_CY}wheel${_RS}${_NL}$(bliner)\n ${_RS}Re-map them recursively to
match destination${_NL} directory-tree, ${_MG}including subfolders${_NL}$(gap)\n ${_RS}e.g In ${_WH}${_AR}Stash_${_RS}
(will sync to ${_MG}/private/var/${_NL}/stash/Applications${_RS} as per above) upload\n Icons for each application to
customize,${_NL} i.e '${_WH}iExample1${_BL}.app${_RS}', into a folder\n With same/matching basename e.${_BK}g
'${_WH}${_AC}iExample1${_RS}' or${_NL} ${_CY}iBlacklist${_RS} or ${_CY}iFile${_RS} or ${_CY}iMobileSafari${_RS}${_NL}$(gap)\n
${_YL}Note: ${_MG}LEAVE OUT${_YL} all directory suffixes ${_RS}i.e ${_NL}Remove '${_RD}.app${_RS}' & '${_RD}.bundle${_RS}' &
'${_RD}.framework${_RS}'${_NL}$(gap)\n In ${_WH}${_AR}Mobile_ ${_RS}copy icons into folders named for${_NL} any
${_WH}AppStore${_BL}.app${_RS} selected for theming;${_NL}$(gap)\n In ${_WH}${_AR}Library_ ${_RS} customize
${_CY}PreferenceLoader${_RS}, ${_NL}${_CY}Activator${_RS}, ${_CY}PreferenceBundles${_RS},
${_CY}ActionMenu${_RS};${_NL}$(gap)\n ${_RS}In ${_WH}${_AR}System_ ${_RS} customize SpringBoard UI images in${_NL}
${_CY}CoreServices${_BL}/${_CY}SpringBoard${_BL}.app${_RS}\n & Many others in ${_CY}PreferenceBundles${_RS},
${_CY}${_NL}Frameworks${_RS}, ${_BK}${_CY}WeeAppPlugins${_RS}, ${_CY}PrivateFrameworks${_RS}${_NL}$(gap);\n The
```

```
script will sync all your icons to their proper destination on condition that:$_NL}$gap)\n $_RD}A/$_RS}They
$_WH}ALSO$(repeat 5 echo -ne $_BK}) EXIST$_RS} in destination $_MG}folder$_CY}/$_MG}subfolder\t$_NL}$$_RS}i.e
$_BL}$$_AC}No new files created$_NL}$gap)\n $_RD}B/$_RS}They have $_WH}DIFFERENT$_MG} Modification Date
$_BK}$$_CY}& $_BK}$$_MG}Size \t$_NL}$$_RS}i.e $_BL}$$_AC}No repeat sync$_NL}$gap)\n $_RS}Application updates
change all modification $_NL}dates; icons are then re-synced$_NL}$gap)\n $_YL}This makes it possible to manually
theme $_NL}iDevice $_CY}easily $_YL}and $_CY}without $_YL}WinterBoard\n $_YL}$$_AC}Getting rid of WinterBoard can
speed up $_NL}iDevices performance noticeably$(liner)\n $_RS}---$_RD}$$_4LINE}$$(repeat 7 printf
"\b")$_RS}HIT$_RD}-$_WH}ENTER$_RD}-$_RS}TO$_RD}-$_RS}CONTINUE$_RD}$$_4LINE}$$(repeat 3 printf
"\b")$_RS}---$(liner)\n"
```

```
#-----
function sfix { notab echo -e "\n $_MG}+$_RS}$$_BL}png$_RS} $_NL}
$_MG}#$_RS}$$_BL}jpg$_RS}-$_BL}gif$_RS}-$_BL}tiff$_RS}-$_BL}psd$_RS} $_NL}
$_MG}*$_RS}$$_BL}plist$_RS}-$_BL}strings$_RS}-$_BL}nib$_RS} $_NL} $_MG}~$_RS}$$_BL}~iphone$_RS} $_NL}
$_CY}i5™$_RS}$$_BL}568h$_RS}"; }
```

```
function bname() { "$@"|sed -e 's:%:.' -e 's:ipod.:ipod-' -e 's: :g' -e 's:\([_\.]*\)*(Pro\)*\.app:.app:' -e
's:\(com\.|libactivator\.|erence\|\.webplugin\|rame\|uration\|tachmen\|sista\|istration\|ting\|ervi\|ication\|Plus\|
Universal\|(\.)*Toggle\|(\.)*Bundle\|.iPhone\|-\.[4}\|-\.[4}\|-\.[4}\|-\.[12}\|):gl' -e
's:\([_\.]*\)*(\.[_\.]*\)|\1\2\3' -e 's:\([_\.]*\10}\|..*\(\.\.\)\|1\2\3/' ; }
```

```
function shortpath() { "$@"|sed -e 's:^.%:.' -e 's:.iphone\.png:%~:' -e 's:\.\/(gl)*png:%+:' -e 's:\.\/(jpg|gif|psd|tiff):%#:' -e
's:\.\/(plist|nib|strings):%*:' -e 's:\.\/(html|css|xml):%^:' -e 's:\-568h(@2x)*:=' -e 's:\.\/(\{13}\|)\.*(\.\.\.\/)\1\2\3/'|sort
-f|awk -F '%' '{printf "\033[34m%05d\033[35m%.1s\033[0m%-18.18s\n",NR,$2,$1;}'|sed -e's/\[34m../[34m/' -e
"s:=-$(cyan)i5™:g"|pa2ste|noindentd sed 's/^/ /'; }
```

```
#-----
function exitxt() { txt;exit; };find /tmp/ -iname rm* |& while read bb;do cat "$bb" >> $_LOGS}/log.previous.log ;rm -f
"$bb";done;cleanup;clear
```

```
#----- Error if not logged in as 'root' -----
[ $(id -u) != 0 ] && exec echo -e "$_RD}\n\n\n ***** $_WH}sorry$_RD} ****$_WH} you are not 'root' $_RD}*****
$_WH}exit$_RD} *****\n\n\n$_RS}" && exitxt
```

```
#-----
##### INSTALL MISSING TOOLS WITH APT-GET #####
```

```
if [[ ! -e /usr/bin/wget ]] || [[ ! -e /usr/bin/plutil ]] || [[ ! -e /usr/bin/awk ]] || [[ ! -e /bin/grep ]] || [[ ! -e /usr/bin/gettext ]]
; then
notab echo -ne "$_RD} Installing necessary tools $_RS}"; etcn
```

```
if [[ ! -e /usr/bin/apt-get ]] ; then
notab echo -e "$_RS} Unable to automatically install tools, please install $_RD}APT 0.7 Strict $_RS}from Cydia." ; exitxt ; fi
if [[ ! -e /private/var/stash/bin/plutil ]] ; then apt-get install com.ericasadun.utilities
if [[ ! $? == 0 ]] ; then notab echo -e "$_RD} Unable to install tools. Are you connected to the Internet?" ; fi ; fi
apt-get install bigbosshackertools
if [[ ! $? == 0 ]] ; then notab echo -e "$_BL} Unable to install tools. Are you connected to the Internet?" ; fi
notab echo -e "$_RD} Done." ; fi
# apt-get -f install
```

```
#-----
##### BEGIN #####
```

```
notab echo -e "\n $_RS}---$_RD}-----$_WH}START$_RD}-----$_RS}---"
nocolor ; repeat 2 dot ; white ; repeat 2 dot ; green ; repeat 2 dot ; yellow ; repeat 2 dot ; magenta ; repeat 2 dot ; red ;
repeat 2 dot ; blue ; repeat 2 dot ; cyan ; repeat 2 dot
notab echo -ne "\n" $_LS}$gap)$$_RD}WARNING!$_YL}REMOVES$_WH}RETINA™$_BL}$*$_CY}!@2x.PNG$_RS} IMAGES
$_NL}ON $_WH}2G$_RS}, $_WH}3G$_RS} & $_WH}3Gs $_RS}DEVICES\n
$_RD}WARNING!$_YL}REMOVES$_WH}REGULAR$_BL}$*$_CY}NON-RETINA™$_BL}$*$_RS}IMAGES $_NL}ON
$_WH}4G$_RS} & $_WH}4Gs $_RS} DEVICES" ; sleep 0 ; printf "\b%.0s" {1..38} ; printf "%.0s" {1..38} ; printf "\b%.0s"
{1..38} ; sleep 0 ; echo -ne "$_CY}UNLESS $_WH}NO $_RD}$*$_CY}RETINA™$_RD}$$_RS} COUNTERPART EXISTS" ; sleep 0 ;
white ; echo -e "$_NL}$gap)"; repeat 5 dot
```

```
if [ -s $_LOGS}/rm.settings.log ] ; then
notab echo -ne "\n $_RD}Use $_MG}script settings $_RD}from prior run $_RS}"; repeat 17 printf '!'; etc ; yn ; answer
if [[ "$ii" = Y ]] ; then printf " $_RS}$$_GR}USE$_WH}CUSTOM$_RS}" ; sleep 1
notab echo -ne "\n\n $_BL}+++++\n
+++++\n $_CY}RECAP$_YL}SETTINGS$_BL}+++++\n ++++++\n"
for E in $(cat "$_LOGS}/rm.settings.log" 2>/dev/null) ; do blue ; repeat 32 printf "+" ; echo -ne "\r"
case "$E" in
```

```
*YES*) notab printf " $_BL}+++++\$_GR}$E$_RS}" | xargs -0 | awk -F '_' '{print $1"\033[37m"$2}';
*NOT*) notab printf " $_BL}+++++\$_RD}$E$_RS}" | xargs -0 | awk -F '_' '{print $1"\033[35m"$2}';
esac ; done ; printf " $_BL}+++++\n"
else printf " $_RS}$$_RD}NEW$_WH}SETTINGS$_RS}" ; sleep 1
rm -f $_LOGS}/rm.settings.log ; fi ; fi
if [[ ! -e $_LOGS}/rm.settings.log ]] ; then
notab printf "\n\n $_RS}++$_LS}$$_YL}CUSTOMIZES$_CY}SCRIPT $_RS}$$_AB}SELECT OPTIONS$_RS}$$_NL}" ; gap
```

```

#-----
##### IDENTIFY DEVICE #####
#-----
notab echo -ne "\n +++${_LS}${_RD}IDENTIFY${_MG}${_AY}i${_WH}${_AC}DEVICES${_RS} ${_AB}by type 2|3|4|5G(s)\t
${_NL}${_RS}${_YL}2${_RS} or [${_CY}3${_RS}] or [${_MG}4${_RS}] or [${_RD}5${_RS}] " ; etc ; ask
while read -s -n 1 _GEN ; do
if [[ $_GEN == "2" || $_GEN == "3" || $_GEN == "4" || $_GEN == "5" ]] ; then break ; fi ; erata ; done
#for _GEN in ${dd} ; do
case "${_GEN}" in
2) _RMIMG="2xPNG";;
3) _RMIMG="2xPNG";;
4) _RMIMG="RMPNG";;
5) _RMIMG="RMPNG";;
esac
printf "${_gap} ${_LS}${_RD}${_AC}i${_GEN}G${_WH}${_RMIMG}${_RS}${_NL}" ; gap
export _RMIMG
#done
#-----
##### SELECT FIRMWARE #####
#-----
notab echo -ne "\n +++${_LS}${_RD}FIRMWARE${_WH}${_AC}SELECT${_RS} ${_AB}iOS Firmware version\t
${_NL}${_RS}${_YL}3${_RS} or [${_CY}4${_RS}] or [${_MG}5${_RS}] or [${_RD}6${_RS}] " ; etc ; ask
while read -s -n 1 fw ; do
if [[ $fw == "3" || $fw == "4" || $fw == "5" || $fw == "6" ]] ; then break ; fi ; erata ; done
_FW=$fw
export _FW
echo -e "i${_GEN}G_FW${_FW} | grep '!' >> ${_LOGS}/rm.settings.log
_RMFW=_RMFW${_FW}
printf "${_gap} ${_LS}${_RD}${_AC}FW${_FW}${_WH}RMFW${_FW}${_RS}${_NL}" ; gap
#-----
##### CUSTOMIZE SCRIPT #####
#-----
declare -a _SET
_SET[0]="\n:+++${_LS}${_RD}F1RStRUN${_WH}${_AC}FREEUP:${_RS}${_AB}Space:on:System:Partition:\n:${_RS}+++${_LS}${_AB}
+++${_BL}${_AM}MOVES${_CY}SYSTEM:${_RS}${_AB}Fonts:&:Audio:to:${_BK}${_BK}${_BK}other:partition\n:${_RS}+++${_LS}${
_AB}+${_BL}${_AM}DISABLES${_CY}UNUSED:${_RS}${_AB}Launch:Daemons\t\t\t"
_SET[1]="\n:+++${_LS}${_RD}SYMLINKS${_WH}${_AC}CREATE:${_RS}${_AB}Useful:Symlinks\t\t\t"
_SET[2]="\n:+++${_LS}${_RD}APT-GET${_WH}${_AC}UPDATE:${_RS}${_AB}Update:hacktools:with:Apt-Get?$(repeat 5 printf
"$_SP")\t"
_SET[3]="\n:${_LS}${_RD}ICONOGRAPHY${_WH}${_AC}MANUAL:${_RS}${_AB}iDevice:$(repeat 8 printf "$_BK")Theme:$(repeat 6
printf "$_BK")Sync:Icons:&:UI:images\t"
_SET[4]="\n:+++${_LS}${_RD}RMEXTRA${_WH}${_AC}REMOVE:${_RS}${_AB}Unused:$(repeat 7 printf
"$_BK")LangPacks,Keyboards,Carriers\t"
_SET[5]="\n:+++${_LS}${_RD}RMSOCIAL${_WH}${_AC}REMOVE:${_RS}${_AB}FaceBook,Twitter,P131,Three20?\t"
_SET[6]="\n:${_LS}${_RD}PERMISSIONS${_WH}${_AC}REPAIR:${_RS}${_AB}Ownerships:&:Permissions\t\t"
_SET[7]="\n:+++++${_LS}${_RD}"$_RMIMG"${_WH}${_AC}REMOVE:${_RS}${_AB}UI:Images:useless:to:${_WH}${_GEN}G${_RS}${
_AB}:devices\t"
_SET[8]="\n:+++++${_LS}${_RD}RMIPAD${_WH}${_AC}REMOVE:${_RS}${_AB}iPad:specific:files?\t\t\t"
#_COUNT="#${_SET[@]}"
_INDEX=(FIRSTRUN SYMLINKS DEBIAN ICONOGRAPHY RMEXTRA RMSOCIAL PERMISSIONS "$_RMIMG" RMIPAD RMIPH5)
for E in ${!_SET[@]} ; do
notab echo -ne "${_SET[E]}" | tr [:] [\ ] ; yn
answer
case "${ii}" in
Y) _YN="YES";;
N) _YN="NOT";;
esac
echo -e "${_YN}_${_INDEX[E]}" | sed 's/DL/RM/' >> ${_LOGS}/rm.settings.log
if [[ $_YN == "YES" ]] ; then
_CHOICE="${_GR}YES${_WH}${_INDEX[E]}${_RS}"
elif [[ $_YN == "NOT" ]] ; then
_CHOICE="${_RD}${_AC}NOT${_RS}${_INDEX[E]}"
fi ; printf "${_gap} ${_LS}${_CHOICES}${_RS}${_NL}" ; gap ; done
#-----
notab echo -ne "\n ${_RS}+++++${_BL}SAVE${_CY}SCRIPT${_RS} settings for next time ${_RS}" ; yn ; answer
if [[ "$ii" = Y ]] ; then printf " ${_GR}YES${_WH}SAVE\n" ; else printf " ${_RD}NOT${_RS}SAVE\n" ; _REMLOG="YES" ; fi
#-----
fi

```

```

#-----
##### RECAP CONFIRM #####
#-----
#notab echo -ne "\n\n  ${_YL}SELECTION${_WH}RECAP\n\n"
#for E in $(cat "${_LOGS}/rm.settings.log" | tee ${_LOGS}/rm.settings.log) ; do
#case "$E" in
#*YES*) notab echo -ne " \t${_RS}----${_GR}${E}" | xargs -0 | awk -F ' ' '{print $1"\033[37m"$2}';
#*NOT*) notab echo -ne " \t${_RS}----${_RD}${E}" | xargs -0 | awk -F ' ' '{print $1"\033[34m"$2}';
#esac ; done
#-----
##### PRE-CUSTOMIZED THE SCRIPT? #####
#-----
notab echo -ne "\n ${_LS}${_RS}${_YL}R${_RS}${_YL}RUN ${_WH}Launch customized script\n ${_RS}${_CY}${_RS}${_CY}SIM
${_WH}Run script as simulation\n ${_RS}${_RD}X${_RS}${_RD}LOG ${_WH}Save settings and cancel${_RS} " ; repeat 16 printf
'.' ; printf " ${_NL}${_RS}${_YL}R${_RS}${_SP}${_SP} or ${_SP}${_CY}${_RS}${_SP}${_SP} or ${_SP}${_SP}${_RD}X${_RS} " ; etc ;
ask
while read -s -n 1 yy ; do yy=${yy^^}
if [[ $yy == "X" || $yy == "S" || $yy == "R" ]] ; then break ; fi ; erata ; done
#-----
#----- RUN the script for real if confirmed -----
#-----
if [[ $yy == 'R' ]] ; then _CUSTOM="YES" ; printf "$(gap) ${_LS}${_YL}RUN${_WH}SCRIPT${_NL}" ; gap
#-----
#----- RUN Simulation -----
#-----
elif [[ $yy = 'S' ]] ; then _CUSTOM="SIM" ; printf "$(gap) ${_LS}${_CY}RUN${_WH}SIM${_NL}" ; gap
#-----
#----- OR CANCEL and EXIT script. Run it again anytime -----
#-----
else _CUSTOM="NOT" ; printf "$(gap) ${_LS}${_RD}END${_WH}EXIT${_NL}" ; gap
echo ; nocolor ; repeat 2 dot ; white ; repeat 2 dot ; green ; repeat 2 dot ; yellow ; repeat 2 dot ; magenta ; repeat 2 dot ; red
; repeat 2 dot ; blue ; repeat 2 dot ; cyan ; repeat 2 dot ; notab echo -e "\n
${_RS}---${_RD}-----${_WH}EXIT${_RD}---${_WH}STAGE${_RD}---${_WH}LEFT${_RD}-----${_RS}---\n\n" &&
exitxt ; fi
#-----
#----- START! -----
#-----
echo ; nocolor ; repeat 2 dot ; white ; repeat 2 dot ; cyan ; repeat 2 dot ; green ; repeat 2 dot ; yellow ; repeat 2 dot ;
magenta ; repeat 2 dot ; red ; repeat 2 dot ; blue ; repeat 2 dot ; liner ; notab echo -ne "\n ${_CY}BEGIN ${_NL}${_RS}PRESS
'${_RD}q${_RS}' at any time to ${_CY}CANCEL PROCESS ${_RS}" ; etcn ; liner ; echo ; nocolor ; dot ; white ; dot ; cyan ; dot ;
green ; dot ; yellow ; dot ; magenta ; dot ; red ; dot ; blue ; dot
#-----
##### FIRST RUN - REMOVE UNUSED SYSTEM FILES, DAEMONS #####
#-----
if [[ "$(grep 'FIRSTRUN' ${_LOGS}/rm.settings.log | awk -F ' ' '{printf $1}')" == "YES" ]] ; then
#-----
#----- 6 50 90 102 63 100 56 -----
#-----
#----- Carrier bundles - Removes all except ((( AT&T , Unknown , CarrierLab , French & Czech ))) bundles -----
#-----
liner ; cleanup ; _RMCB="Y"
notab echo -ne "\n ${_RD}Searching for ${_WH}Carrier Bundles${_RD} to remove${_RS} " ; etcn
if grep -q '^_XSIM' ${_LOGS}/rm.logxcept.log 2>/dev/null ; then
notab echo -ne "\n ${_RS}Some ${_LS}${_YL}Carrier Bundles${_RS} have already been removed ${_YL}:: ${_NL}${_WH}Remove
some more ${_RS}" ; etc ; yn ; answer ; print_yn ; if [[ "$ii" = Y ]] ; then _RMCB="Y"
else _RMCB="N" ; notab echo -e "\n ${_MG}Cancelling ${_WH}Carrier Bundles${_MG} search " ; fi ; fi
if [[ "$_RMCB" = Y ]] ; then blank
if grep -q '^_XSIM' ${_LOGS}/rm.logxcept.log 2>/dev/null ; then
notab echo -ne "\n ${_RS}${_GR}0${_RS}${_WH}REAPPLY ${_RS}previous ${_RD}Carrier Bundle${_RS} exceptions${_BL} or\n
${_RS}${_RD}1${_RS}${_WH}DISCARD ${_RS}exceptions ${_BL}&${_RS} find ${_WH}all${_RD} Carrier Bundles ${_RS}" ; etc ; oi ;
zanswer
print_oi ; sleep 1 ; if [[ "$ff" = 0 ]] ; then
_XSIM="$(grep '^_XSIM' ${_LOGS}/rm.logxcept.log 2>/dev/null | tr '|' '\n' | sed -e 's:^\|<:' -e 's:$\|>:' | grep -v '_XSIM' | tr
'\n' '|')" ; fi ; fi
notab echo -e "\n ${_CY}Listing ${_WH}CARRIERS ${_CY}first ${_RS}::${_CY} then ${_WH}BUNDLES ${_RS}\n"
find "/System/Library/Carrier Bundles" -iname '*.bundle' -type d | grep -Ew "$_XSIM)_XSIM" | grep -Ev
'ATT_US|Unknown|CarrierLab|OQCFactoryUMTS' | tee -a /tmp/tf.logrm.log | sed -e 's:.,*/,' -e 's:\.bundle:.' | cat -n | sed
's:^\| \([0-9]*\)\| \[t\]: \1:g' | tee -a /tmp/tf.logcarrier.log | sed -e 's: [0-9][0-9]*[0-9]*:.' -e 's:_.*:' | tr 'a-z' 'A-Z' | sort -fu

```

```

| tee -a /tmp/tf.carriers.log | awk '{printf "\033[31m%03d\033[0m%-10.10s\n",NR,$1}' | pa4ste | sed 's:^\: '
if [ -s /tmp/tf.logrm.log ] ; then
notab echo -ne "\n ${_YL}You can ${_WH}edit${_YL} the list to keep ${_WH}selected${_YL} Carrier Bundles ${_RS}" ; etcnn
notab echo -ne " Do you want to ${_CY}EDIT${_RS} the ${_WH}Carrier Bundle${_RS} removal list ${_NL}" ; etc ; yn ; answer ;
print_yn ; if [[ "$ii" = Y ]] ; then
notab echo -ne "\033[1A ${_RD}DEFINE EXCEPTIONS ${_RS}:: First select the ${_WH}Carriers${_RS} :: then edit the
${_WH}Bundles${_RS} list\n enter ${_CY}number${_RS}(${_CY}${_BK}s${_RS})${_BK}${_RS} ${_RS}separated by a ${_WH}single
space ${_RS}& hit [${_MG}RETURN${_RS}]${_YL}\n\n " ; txt && read _CARRIER ; if [[ $_CARRIER = "" ]] ; then notab printf
"\033[1A\033[1A" ; blank ; notab echo -e "\n No selections"
else for _CR in $_CARRIER ; do rm -f /tmp/tf.tmp_sim*.log ; bliner
notab echo -e "\n ${_MG}CARRIER${_RS}: ${_BL}$(sed -n "${_CR}p" /tmp/tf.carriers.log)\n"
grep -Ei "$(sed -n "${_CR}p" /tmp/tf.carriers.log)" /tmp/tf.logcarrier.log | sed 's:([0-9][0-9]*[0-9]*)[a-zA-Z0-9]*_(.*):\1
\2:' | tr 'a-z' 'A-Z' |& while read _CRC ; do
echo $_SIMS | tr ' ' '\n' | grep -Ei "^$(echo ${_CRC} | sed 's:^\: \\\:)' | sed 's:^\:$(echo ${_CRC} | grep -o
'[a-zA-Z0-9]*_):'" | sed -e "s:^\:$_CRC%*" :>> /tmp/tf.tmp_sim.log ; done
if (( "$wc -l < /tmp/tf.tmp_sim.log" == "1" )) ; then notab echo -ne "${_RD}Take this ${_WH}Carrier ${_RS}off the Removal
List " ; etc ; yn ; answer ; print_yn ; if [[ "$ii" = Y ]] ; then
notab echo -e "\033[1A$(blank) ${_RS}]${_CY}✓${_RS}] MARKED for ${_YL}KEEPING" ; cat /tmp/tf.tmp_sim.log >>
/tmp/tf.tmp_XKEEP_sim.log ; else printf "\033[1A" ; fi ; else
cat /tmp/tf.tmp_sim.log | tee /tmp/tf.tmp_sim_all.log | sed 's:^\[0-9]* ::g' | tr 'a-z' '%A-Z' | cat -n | awk '{printf
"\033[36m%02d\033[0m%-16.16s\n",$1,$2}' | tr '%' ' ' | pa2ste | sed -e 's:^\: ' ; _TLOG="tmp_sim" ; editlist
_PRINT="$(sed 's:^\[0-9]* ::g' /tmp/tf.tmp_sim.log | tr 'a-z' '%A-Z' | cat -n | awk '{printf
"\033[36m%02d\033[0m%-16.16s\n",$1,$2}' | tr '%' ' ' | pa2ste | sed 's:/ /' ; printf "\033[1A)" ; editprint
grep -Ev "$(cat /tmp/tf.tmp_sim.log)" /tmp/tf.tmp_sim_all.log >> /tmp/tf.tmp_XKEEP_sim.log ; fi
done ; fi ; _SIMKEEP="$(awk '{printf "%sp;",$1}' /tmp/tf.tmp_XKEEP_sim.log 2>/dev/null | sed 's;/;/)' ; fi
sed -n "${_SIMKEEP}" /tmp/tf.logrm.log > /tmp/tf.carrierkeep.log
grep -Ev "_SIMKEEP$(cat /tmp/tf.carrierkeep.log 2>/dev/null)" /tmp/tf.logrm.log >> /tmp/rm.carrier.log
cat /tmp/rm.carrier.log > /tmp/tf.logrm.log ; bliner
notab echo -e "\n ${_CY}Found ${_RS}&${_CY} logged [${_RS}$(icount)${_CY}] ${_WH}Carrier Bundles${_CY} to remove"
cat /tmp/tf.carrierkeep.log 2>/dev/null | sed -e 's:.,*/,' -e 's:.\:*\: ' > /tmp/tf.1.log
grep '^_XSIM' ${_LOGS}/rm.logxcept.log 2>/dev/null | sed 's:|\n:g' | grep -v -e '_XSIM' >> /tmp/tf.1.log
sort -fu /tmp/tf.1.log | sed -e 's:^\: ' | tr -d '\n' | sed -e 's:^\: _XSIM|:' -e 's:$_\n:' > /tmp/tf.2.log
grep -v '^_XSIM' ${_LOGS}/rm.logxcept.log 2>/dev/null >> /tmp/tf.2.log
cat /tmp/tf.2.log > ${_LOGS}/rm.logxcept.log
notab echo -ne "\n ${_RD}Save ${_WH}Carrier Bundles${_RD} Remove-list to
${_MG}mobile${_RD}:${_MG}Documents${_RD}:${_MG}Logs ${_RS}" ; etc ; yn ; answer
print_yn ; if [[ "$ii" = Y ]] ; then cat /tmp/rm.carrier.log > ${_LOGS}/rm.carrier.log ; fi ; else
notab echo -ne "\n ${_RS}Found no ${_CY}Carrier Bundles${_RS} to remove" ; etcn ; fi ; fi
#-----
#----- TextInput bundle removal ; default rm all except: english, japanese (emoji) -----
#-----
linen ; cleanup
notab echo -ne "\n ${_RD}Searching for unused ${_WH}TextInput${_RD} bundles ${_RS}" ; etcn
if [[ $(grep '^_XLANG' ${_LOGS}/rm.logxcept.log 2>/dev/null) ]] ; then
notab echo -ne "\n ${_CY}A previous ${_WH}exclusions logfile ${_CY}was found " ; cutxt "on yr iDevice ${_RS}" ; etcn
notab echo -ne "\n ${_GR}RE-USE${_RS}]${_GR}0${_RS}]${_WH}old ${_RS}file ${_BL}or\n
${_RD}DEFINE${_RS}]${_RD}1${_RS}]${_WH}new ${_RS}file ${_AB}.${_SP}" ; cutxt '.....' ; etc ; oi ; zanswer
print_oi ; sleep 1 ; if [[ "$fff" = 1 ]] ; then
grep -v '^_XLANG' ${_LOGS}/rm.logxcept.log > /tmp/tf.x2.log
cat /tmp/tf.x2.log > ${_LOGS}/rm.logxcept.log ; fi ; fi
if [[ ! $(grep '^_XLANG' ${_LOGS}/rm.logxcept.log 2>/dev/null) ]] ; then
notab echo -ne "\n ${_RS}First enter ${_CY}TextInput Language${_RS}(${_CY}${_BK}s${_RS})${_BK}${_RS} to
${_WH}exclude${_RS}$(cutxt ' from search'), ${_NL}${_RS}Separated by a ${_WH}single space ${_RS}& hit
[${_MG}ENTER${_RS}]${_YL}\n\n " ; txt && read _XCLUDE
for X1 in $_XCLUDE ; do
while [[ ! $(echo ${_LCODES} | sed -e 's:^\: \n:g' | grep -iw "^${X1}") ]] ; do
notab echo -ne "\n ${_WH}${X1} ${_RD}is not a known ${_WH}Keyboard Language${_RS} ... \n\n ${_RS}Type it again or leave
empty to escape & hit [${_CY}ENTER${_RS}] ${_YL}" ; txt && read X1
if [[ ! $X1 ]] ; then break ; fi ; done
echo $_LCODES | sed -e 's:^\: \n:g' | grep -iw "^${X1}" >> /tmp/tf.tmp.log ; done
sed 's:^\: ' /tmp/tf.tmp.log | tr -d '\n' | sed -e 's:^\: _XLANG:' -e 's:$_\n:' >> ${_LOGS}/rm.logxcept.log
notxt ; fi ; echo
_XCLTXT="$(grep '^_XLANG' ${_LOGS}/rm.logxcept.log 2>/dev/null | tr '|' '\n' | grep '^[a-z]{2}$' | tr '\n' '|' | sed -e
's:|\.\.bundle|:g)"
find ${_SYSLIB}/TextInput -iname '*.bundle' -type d | grep -Eiv "${_XCLTXT}_en\.\_emoji" > /tmp/tf.logrm.log
echo $_LCODES | sed 's:^\: \n:g' | grep -Eiw "$(sed -e 's:(^\: \.\.bundle)::g' -e 's:^\: \<:g' -e 's:^\: \>:' /tmp/tf.logrm.log)" |
awk -F '|' '{printf "%s\n",$1}' | sort | tee /tmp/tf.txtinput.log | awk '{printf "\033[36m%02d\033[0m%-10s\n",NR,$1}' |
pa3ste

```

```
if [ -s /tmp/tf.txtinput.log ] ; then _TLOG="txtinput" ; editlist ; _PRINT=$(awk '{printf "\033[36m%02d\033[0m%-10s\n",NR,$1}' /tmp/tf.txtinput.log | pa3ste
printf "\033[1A\033[1A") ; editprint ; _XCPT="LANG" ; logxlang
_XCLTXT2=$(grep '^_XLANG' ${LOGS}/rm.logxcept.log 2>/dev/null | tr '|' '\n' | grep '^([a-z]){2}$' | tr '\n' '|' | sed -e
's:|\\\.bundle|:g' -e 's:|:|:')
grep -Fw "${XCLTXT2}" /tmp/tf.logrm.log | tee /tmp/tf.logtmp.log > /tmp/rm.txtinput.log
cat /tmp/tf.logtmp.log > /tmp/tf.logrm.log
notab echo -e "\n$(blank) ${CY}Found ${RS}&${CY} logged [${RS}$(icount)${CY}] ${WH}TextInput${CY} bundles in
iDevice"
else notab echo -e " ${BL}Found no ${WH}TextInput${BL} bundles to remove" ; fi
#-----
#----- Remove unnecessary keyboard layouts -----
#-----
liner ; cleanup
_iKB='KeyboardLayouts/USBKeyboardLayouts.bundle/uchrs Frameworks/UIKit.framework'
notab echo -ne "\n ${RD}Searching for unused ${WH}Keyboard Layouts${RD}" ; cutxt "to remove${RS}" ; etcn
if [[ ! $(grep '^_XLANG' ${LOGS}/rm.logxcept.log 2>/dev/null) ]] ; then
notab echo -ne "\n\n ${RS}First enter ${CY}Keyboard Language${RS}(${CY}${BK}s${RS})${BK}${RS} to
${WH}exclude${RS} from search, ${NL}${RS}separated by a ${WH}single space ${RS}& hit
[${MG}ENTER${RS}]${YL}\n\n" ; txt && read _YCLUDE
for Y1 in $_YCLUDE ; do
while [[ ! $(echo ${LCODES} | sed -e 's:%:\n:g' | grep -i "^${Y1}") ]] ; do
notab echo -ne "\n ${WH}${Y1} ${RD}is not a known ${WH}Keyboard Language${RS} ... \n\n ${RS}Type it again or leave
empty to escape & hit [${CY}ENTER${RS}] ${YL}" ; txt && read Y1
if [[ ! ${Y1} ]] ; then break ; fi ; done
echo ${LCODES} | sed -e 's:%:\n:g' | grep -iw "^${Y1}" | tr '\n' '|' >> /tmp/tf.tmp.log ; done ; notxt
cat /tmp/tf.tmp.log 2>/dev/null | sed "s:^\$(grep '^_XLANG' ${LOGS}/rm.logxcept.log 2>/dev/null):" | sed -e
's:|([A-Z][^]*)\n:1:g' -e 's:|:|' | grep -v '^_XLANG' | sort -u | tr '\n' '|' | sed 's:^\_XLANG|:' >> /tmp/tf.tmp1.log
grep -v '^_XLANG' ${LOGS}/rm.logxcept.log | sed 's:^\n: 2>/dev/null >> /tmp/tf.tmp1.log
grep '!' /tmp/tf.tmp1.log > ${LOGS}/rm.logxcept.log && rm -f /tmp/tf.tmp*.log ; fi
_XCL="$(grep '^_XLANG' ${LOGS}/rm.logxcept.log 2>/dev/null | tr '|' '\n' | sed -e 's:^\([a-z]*\)s:\1\\.plist:' | grep -v
'_XLANG' | tr '\n' '|")
for E in $_iKB ; do cleanup ; function fn() { printf "Library/${E}" ; }
find ${SYSLIB}/${E} -type f | grep -E 'Keyboard-|\.kbstar|\.uchr' | grep -Eiv
"${XCL}Common|default|English|emoji|intl|5Row|Latin|Control|NumberPads|Dvorak|UnicodeHexInput|US|.uchr|USExtended
|USInternationalPC|USBKeyboardLayouts|.plist|en\_US|Info|.plist|en|.plist|\-chr" >> /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then echo
if grep -q 'uchr' /tmp/tf.logrm.log ; then
sed -e 's:./,/' -e 's:\.(\.*/)' -e 's/\(\(2Set\)*[A-Z]{1}\)[a-z][a-z]*\)\([A-Z]{1}\.*/\)/1/' /tmp/tf.logrm.log | sort -fu | tee
-a /tmp/tf.uchr.log | awk '{printf "\033[36m%02d\033[0m%-10s\n",NR,$1}' | pa3ste | noindentd sed 's:/ /'
_TLOG="uchr" ; editlist ; _PRINT=$(awk '{printf "\033[36m%02d\033[0m%-10s\n",NR,$1}' /tmp/tf.uchr.log 2>/dev/null |
pa3ste | noindentd sed 's:/ /'
editprint ; _XCPT="LANG" ; logxlang
grep -E "${sed 's:^\n:g' /tmp/tf.uchr.log 2>/dev/null}" /tmp/tf.logrm.log >> /tmp/rm.keyboard.log ; cat
/tmp/rm.keyboard.log 2>/dev/null > /tmp/tf.logrm.log ; _iLCount=$(wc -l < /tmp/tf.uchr.log)
else sed -e 's:./,/' -e 's:\.(\.*/):' -e 's:(Keyboard-|iPhone|iPad):g' /tmp/tf.logrm.log > /tmp/tf.keyboards.log
echo ${LCODES} | sed 's:%:\n:g' | grep -Ei "${cat /tmp/tf.keyboards.log | sed -e
's:|(en|fr|de|es)\([-_])\([a-zA-Z0-9]*\):\2\3:g' -e 's:|([a-zA-Z]{2}\[-_])\.:1:g' -e 's:|_:0:g' -e 's:^\|<:' -e 's:|\|>:'
| uniq) | awk -F '|' '{printf "%s\n", $1}' | tee /tmp/tf.logtmp.log | awk '{printf "\033[36m%02d\033[0m%-10s\n",NR,$1}' |
pa3ste | noindentd sed 's:/ /'
_TLOG="logtmp" ; editlist ; _PRINT=$(awk '{printf "\033[36m%02d\033[0m%-10s\n",NR,$1}' /tmp/tf.logtmp.log
2>/dev/null | pa3ste | noindentd sed 's:/ /'
editprint ; _XCPT="LANG" ; logxlang
grep -Ei "${(echo ${LCODES} | sed -e 's:%:\n:g' | grep -Ei "${cat /tmp/tf.logtmp.log 2>/dev/null}" | sed -e 's:^[a-zA-Z]*::g'
-e 's:0\([a-zA-Z4]\):[-_]\1:g' -e 's:|*:\.plist:g)" /tmp/tf.logrm.log | tee /tmp/tf.logrm2.log >> /tmp/rm.keyboard.log
cat /tmp/tf.logrm2.log 2>/dev/null > /tmp/tf.logrm.log ; _iLCount=$(wc -l < /tmp/tf.logtmp.log) ; fi
notab echo -ne "\n ${RD}Found ${RS}&${RD} logged [${RD}]${WH}$(icount | sed -e 's:$(red)] $(magenta)Keyboard
Layouts:" -e '/^\<[>\<1>\>\<1>\>\<1>\>/s/Layouts/Layout/'${RD} $(allcaps 'f')or total [${RD}]${WH}$_iLCount${RD}]
${MG}distinct languages${RD} in${NL} ${WH}$(hobble shortd fn)\n" | sed -e 's:\.bund\(\le\)*::' -e
'/[^\0-9]1[^\0-9]/s/languages/language/' -e 's:/:$(red)\:$(white):g"
else notab echo -e "\n ${BL}No unused [${RS}]${AM}Keyboard Layouts [${BL}]in [${NL}]${WH}System${RD}:${WH}$(hobble
shortd fn)" | sed -e 's:\.bund\(\le\)*::' -e 's:/:$(red)\:$(white):g"
fi ; done
#-----
#----- Remove AdSheet Demo FieldTest GameCenter Diagnostics Setup Stocks MobileStore apps -----
#-----
liner ; cleanup ; notab echo -ne "\n ${RD}Searching for [${WH}]Stock Applications${RD}" ; cutxt "to remove${RS}" ; etcn
_APP2RM="AdSheet DemoApp FieldTest Game iOS MobileStore Nike Setup Stocks"
```



```

_COUNT=0
_XREM="$ (grep '^_XREM' ${LOGS}/rm.logxcept.log 2>/dev/null | sed -e 's:_XREM|::')"
_XKEEP="$ (grep '^_XKEEP' ${LOGS}/rm.logxcept.log 2>/dev/null | sed 's:^_XKEEP|::' | tr '\n' '\n' | grep -Ew "${_XREM}" | tr '\n' '\n' | sed 's:|:|::')"
for E in $_APP2RM ; do
if [[ -e $(find ${STAPPS}/${E}.app/ -maxdepth 2 | grep -E "${E}.*\.app/${E}[\^\.]*$") ]] ; then
if [[ $(echo $_XKEEP | grep -Eo "${E}") ]] ; then
appname printf "\n ${LS}${MG}Remove existing '${RD}PROTECTED${MG}' status of '${WH}${E}${RS}.app${MG}'${RS} " ;
etc ; yn ; answer ; print_yn ; bottom
if [[ "$ii" = Y ]] ; then blank ; printf "\033[1A" ; _XKEEP=$(printf "$_XKEEP" | tr '\n' '\n' | grep -v "${E}" | sort -fu | tr '\n' '\n' | sed 's:|:|::') ; fi ; _XKEEP=$(xkeep) ; fi
if [[ $(echo $_XREM | grep -Eo "${E}") ]] ; then
appname printf "\n ${MG}Already marked '${RS}${E}${MG}' for removal ${RS}" ; etcn ; sleep 1 ; blank ; bottom
ii="Y" ; elif [[ ! $(echo $_XKEEP | grep -Eo "${E}") ]] ; then printf "\n" ; bottom
notab appname printf " ${LS}${YL}Remove: '${WH}${E}${YL}.app ${RS}" ; etc ; yn ; answer ; print_yn ; fi
if [[ "$ii" = N ]] ; then appname echo -ne "\033[1A$(blank) ${LS}${CY}Skipping ${WH}${E}${RS}.app\n" ; sleep 1
_XKEEP=$(xkeep)
elif [[ "$ii" = Y ]] ; then let _COUNT++ ; _XREM=$(xrem)
appname echo -ne "\033[1A$(blank) ${LS}${CY}Marking ${WH}${E}${RS}.app\n"
if [[ "${E}" = MobileStore ]] ; then
find ${STAPPS}/${E}.app/ -mindepth 1 -maxdepth 1 2>/dev/null | grep -iv 'icon\|English\|Info' >> /tmp/tf.logrm.log
elif [[ "${E}" = Setup ]] ; then echo "${STAPPS}/Setup.app" >> /tmp/tf.logrm.log
echo /System/Library/SetupAssistantBundles >> /tmp/tf.logrm.log
mkdir -p ${DOWNL}/Backup/LaunchDaemons
mv -f ${SYSLIB}/LaunchDaemons/com.apple.SpringBoard.SetupHelper.plist ${DOWNL}/Backup/LaunchDaemons/
2>/dev/null
else find ${STAPPS} -maxdepth 1 | grep -i "${E}" >> /tmp/tf.logrm.log ; fi ; fi ; fi ; done
if [ -s /tmp/tf.logrm.log ] ; then cat /tmp/tf.logrm.log > /tmp/rm.apps.log ; fi
grep -Ev '_XREM|_XKEEP' ${LOGS}/rm.logxcept.log 2>/dev/null > /tmp/tf.x1.log
echo "${_XREM}" | tr '\n' '\n' | sort -u | sed 's:^:|:' | tr -d '\n' | sed -e 's:^(\|):_XREM|1:' -e 's:|:\n:' >> /tmp/tf.x1.log
echo "${_XKEEP}" | tr '\n' '\n' | sort -u | grep -Ew "${_XREM}" | tr '\n' '\n' | sed -e 's:^(\|[A-z]):_XKEEP|1:' -e 's:|:\n:' >>
/tmp/tf.x1.log ; grep '.' /tmp/tf.x1.log > ${LOGS}/rm.logxcept.log 2>/dev/null
if (( _COUNT == 0 )) ; then notab echo -e "\033[1A$(blank) ${BL}Found no ${WH}Stock Applications ${BL}to remove" ;
else
notab echo -ne "\n ${BL}Marked ${RD}${COUNT}${BL} stock applications for removal${RS} " ; etcn ; fi
#-----
#----- Remove AdSheet Braille GameKit Nike Victoria Setup Sports bundles -----
#-----
liner ; cleanup ; notab echo -ne "\n ${RD}Select ${WH}iOS bundles${RD} to mark for removal ${RS}" ; etcn
_iNXS="AdSheet|iAd Braille|ScreenReader Game MobileMe Nike|VictoriaSettings Setup
SportsTrainer|SportsVoices|SportsWorkout Stocks Twitter"
_XREM="$ (grep '^_XREM' ${LOGS}/rm.logxcept.log 2>/dev/null | sed -e 's:_XREM|::')"
_XKEEP="$ (grep '^_XKEEP' ${LOGS}/rm.logxcept.log 2>/dev/null | sed 's:^_XKEEP|::' | tr '\n' '\n' | grep -Ew "${_XREM}" | tr '\n' '\n' | sed 's:|:|::')"
for E in $_iNXS ; do cleanup
if [[ $(appname printf "${SYSLIB}/AccessibilityBundles/${E%|*}.axbundle" | sed -e 's:(Sports[a-zA-Z]*):\1Framework:' -e 's:(MobileMe):PublishTo\1:') ]] ; then
if [[ $(appname echo "${E%|*}" | grep -Eo "${_XREM}") ]] ; then echo
appname notab printf " ${MG}Already marked '${RS}${E%|*}${MG}' for removal ${RS}" ; etcn ; ii="Y" ; sleep 1
else echo ; bottom ; appname notab echo -ne " ${YL}Remove '${RD}${E%|*}${YL}' " ; cutxt "bundles ${RS}"
etc ; yn ; answer ; print_yn ; blank ; fi
if [[ "$ii" = N ]] ; then _XKEEP=$(xkeep) ; appname notab echo -e "\033[1A$(blank) ${LS}${CY}Skipping
${WH}${E%|*}${CY} bundles" ; sleep 2
elif [[ "$ii" = Y ]] ; then _XREM=$(xrem) ; bottom
appname notab printf "\033[1A$(blank) ${LS}${WH}Finding '${YL}${E%|*}${WH}' bundles ${RS}" ; etc ; sleep 1
if [[ "${E}" = 'Game' || "${E}" = 'Twitter' ]] ; then find $Path -iname "*${E}*" -type d | grep -Ev "Caches|${STAPPS}" >>
/tmp/tf.logrm.log
else find ${SYSLIB} -maxdepth 2 | grep -Ei "${E}" >> /tmp/tf.logrm.log
find /Library -maxdepth 2 | grep -Ei "${E}" >> /tmp/tf.logrm.log ; fi
if [ -s /tmp/tf.logrm.log ] ; then
appname notab echo -e "\r ${LS}${CY}Found ${RS}&${CY} logged [${WH}${icount | sed -e 's:|:(cyan)} $(white)${E%|*}
$(cyan)bundles:" -e '/^\<[><1><1><1>>/s/bundles/bundle/' "${RS}"
grep -Ei "${E}" /tmp/tf.logrm.log |& while read _LINE ; do
notab printf "\n $(echo ${_LINE%/} | sed -e 's:(\.[^/]*\ |):' -e 's:([/^\][6\}][^/]*\([^\]\{1\})[^\]:\1^\2:g' -e 's:/:.' -e 's:/:(red):\:(nocolor):g')$(echo ${_LINE##} | sed 's:(\.[^\.])[\^\.]*:\1:')" ; done
echo ; cat /tmp/tf.logrm.log >> /tmp/rm.apps.log
else appname notab echo -e "${blank} ${LS}${BL}Found no ${RS}${E%|*}${BL} bundles" ; fi
else _XREM=$(xrem) ; fi ; fi ; done

```

```

grep -Ev '^_XREM|_XKEEP' ${LOGS}/rm.logxcept.log 2>/dev/null > /tmp/xf.x1.log
xsub echo "${_XREM}" | tr '\n' | sort -u | sed 's:^\(:\):_XREM\1:' -e 's:^\(:\):_XKEEP\1:' -e 's:^\(:\):_XKEEP\1:' -e 's:^\(:\):_XKEEP\1:' >> /tmp/xf.x1.log
xsub echo "${_XKEEP}" | tr '\n' | sort -u | grep -Ew "${_XREM}" | tr '\n' | sed -e 's:^\(:\):_XKEEP\1:' -e 's:^\(:\):_XKEEP\1:' -e 's:^\(:\):_XKEEP\1:' >> /tmp/xf.x1.log
grep '! /tmp/xf.x1.log > ${LOGS}/rm.logxcept.log 2>/dev/null
#----- Removing selected Stock-Apps & Stock-iOS bundles -----
#-----
if [ -s /tmp/rm.apps.log ] ; then bliner
cat /tmp/rm.apps.log > /tmp/xf.apps.log
_INXS='AdSheet|iAd Braille|ScreenReader Game MobileMe Nike|VictoriaSettings Setup
SportsTrainer|SportsVoices|SportsWorkout Stocks Twitter'
_APP2RM='AdSheet DemoApp FieldTest Game iOS MobileStore Nike Setup Stocks'
_XREM=$(grep '^_XREM' ${LOGS}/rm.logxcept.log 2>/dev/null | sed 's:^_XREM|:')
_AX=$(echo "$_INXS $_APP2RM" | tr '\n' | sed -e "s:(iAd):\1\nAdSheet:" -e "s:(ScreenReader):\1\nBraille:" -e
"s:(iOS):\1 Diagnostics:" -e "s:(Victoria)[A-z]*:\1\nNike:" -e "s:(Sports)\(Trainer|Voices|Workout):\1:" | sort -fu | tr
'\n' | sed 's:|:|')
notab echo -e "\n ${_RD}MARKED ${_RS}for ${_YL}REMOVAL${_RS}:\n"
grep -Eo "$_AX" /tmp/xf.apps.log | sort -fu | tee -a /tmp/xf.marked.log | awk '{printf
\033[36m%02d\033[0m%-10s\n,NR,$1}' | sed -e "s:(Game):\1Kit \t\t$(nocolor)\[$(magenta)Game
Center$(nocolor)\]$(cyan):" -e "s:(AdSheet|iAd):AdSheet \t\t$(nocolor)\[$(magenta)iAd Bundles$(nocolor)\]$(cyan):" -e
"s:(Braille|ScreenReader):ScreenReader\t\t$(nocolor)\[$(magenta)Braille Kit$(nocolor)\]$(cyan):" -e
"s:(Nike|Victoria):Nike\t\t$(nocolor)\[$(magenta)&. Victoria$(nocolor)\]$(cyan):" -e
"s:(FieldTest):\1\t\t$(nocolor)\[$(magenta)iBenchmarks$(nocolor)\]$(cyan):" -e
"s:(Sports):\1\t\t$(nocolor)\[$(magenta)Trainer Kit$(nocolor)\]$(cyan):" -e "s:(iOS):iOS Diagnostics $(cyan):"
_TLOG="marked" ; editlist
_XDEL=$(echo $_INXS | tr '\n' | grep -E "$_CC" | sed 's:|.*$:' |& while read E; do xsub echo $E; done | tr '\n' | sed
's:|:|')
_PRINT=$(awk '{printf "\033[36m%02d\033[0m%-10s\n,NR,$1}' /tmp/xf.marked.log) ; editprint
grep '^_XREM' ${LOGS}/rm.logxcept.log 2>/dev/null | sed 's:^_XREM|: | tr '\n' | grep -Ev "$_XDEL" | tr '\n' | sed -e
's:^_XREM|:' -e 's:|:|' > /tmp/xf.x1.log
grep -v '^_XREM' ${LOGS}/rm.logxcept.log 2>/dev/null >> /tmp/xf.x1.log
cat /tmp/xf.x1.log > ${LOGS}/rm.logxcept.log
grep -Fi "$(cat /tmp/xf.marked.log)" /tmp/xf.apps.log > /tmp/rm.apps.log
if [ -s /tmp/rm.apps.log ] ; then
while read -r _LINE ; do rm -fr "$_LINE" ; done < /tmp/rm.apps.log
notab echo -ne "\n$(blank) ${_LS}${_BL}Removed marked ${_CY}Bundles ${_BL}&${_CY} Apps${_BL} from iDevice ${_RS}" ; etcn
; cat /tmp/rm.apps.log > /tmp/rm.removed.log ; rm /tmp/rm.apps.log ; fi ; fi
#----- Remove Nike & Twitter from Settings.plist -----
#-----
bliner ; cleanup
_PLISTW="Nike Twitter"
_XREM=$(grep '^_XREM' ${LOGS}/rm.logxcept.log 2>/dev/null | sed 's:^_XREM|:')
for E in $_PLISTW ; do
if [[ $(echo $_XREM | grep -Eo "$E") ]] ; then
if [[ "$E" = Twitter ]] ; then
rm -fr /private/var/mobile/Library/Twitter
mkdir -p ${_DOWNL}/Backup/LaunchDaemons
chown mobile:mobile ${_DOWNL}/Backup/LaunchDaemons
mv -f ${_SYSLIB}/LaunchDaemons/com.apple.twitter.plist ${_DOWNL}/Backup/LaunchDaemons 2>/dev/null
fi ; _SETTINGS="${_STAPPS}/Preferences.app/Settings.plist"
notab echo -e "\n ${_RD}Removal of ${_WH}$E ${_RD}bundles requires $(allcaps "m")odification of ${_WH}PLIST${_RD} &
${_WH}STRINGS${_RD} text\n"
if grep -q "${E^^}" ${_SETTINGS} 2>/dev/null ; then
plutil -convert xml1 ${_SETTINGS} > /dev/null
tr '\t' '%' < ${_SETTINGS} | sed -e 's:%%<dict>+::g' -e 's:%%</dict>:@:g' > /tmp/Settings1.plist && rm ${_SETTINGS}
tr '\n' '~' < /tmp/Settings1.plist > /tmp/Settings2.plist && rm /tmp/Settings1.plist
sed -e 's:~+[@]*${E^^}[+@]*@: < /tmp/Settings2.plist > /tmp/Settings1.plist && rm /tmp/Settings2.plist
sed -e 's:~+%%<dict>:g' -e 's:@:%%</dict>:g' /tmp/Settings1.plist | tr '~' '\n' > /tmp/Settings2.plist && rm
/tmp/Settings1.plist
tr '% '\t' < /tmp/Settings2.plist > ${_SETTINGS} && rm /tmp/Settings2.plist
plutil -convert binary1 ${_SETTINGS} > /dev/null
chmod 664 ${_SETTINGS}
notab echo -e " ${_CY}Edited ${_WH}$E${_CY} property list ${_WH}Settings${_RS}.plist${_CY} in
${_WH}Preferences${_RS}.app\n"
else notab echo -ne " ${_WH}$E ${_CY}Property list ${_RS}& ${_CY}Localization pack ${_RS}${_AM}$(allcaps "h")ave
${_MG}already${_RS}${_AM} been edited\n" ; fi

```

```

#-----
#----- And from Localization pack -----
#-----
_STR="${_STAPPS}/Preferences.app/*.lproj/Settings*.strings"
for B in $_STR ; do
if grep -Eq "${E^^}" $B ; then
plutil -convert xml1 "$B" > /dev/null
grep -Eiv "${E^^}" < $B > /tmp/Settings.strings && rm $B
cat /tmp/Settings.strings > $B
plutil -convert binary1 $B > /dev/null
chmod 664 $B ; fi ; done
if [ -s /tmp/Settings.strings ] ; then
_LANG="$(echo $_LCODES | sed -e 's:%:\n:g' | grep -Eiw "$(dirname "$B" | sed -e 's:^.*!/:!' -e 's/\./*$//') -e 's:[-_:]0:' -e 's:en(0):\1:' -e 's:\([a-z]\{2\}0\)[A-Z]\{2\}:\1:')" | awk -F '|' '{printf "%s", $1}')"
notab echo -e " ${_CY}Edited ${_WH}${E} ${_CY}localization strings ${_NL}${_WH}$(basename "$B" | sed
"s:\(\(~iphone\)*\(\.\.):\{\$_RS\}\1\2:") ${_CY}in ${_WH}Preferences${_RS}.app\n"
notab echo -ne " ${_BL}${E}${_RS} settings have been removed ${_RS}" ; etcn
rm /tmp/Settings.strings ; fi ; fi ; done
#-----
#----- Modify PLIST in BulletinBoard.framework & com.apple.springboard to reflect removed bundles -----
#-----
cleanup
_PLISTX="Stocks Game"
_XREM=$(grep '^_XREM' ${LOGS}/rm.logxcept.log 2>/dev/null | sed 's:^_XREM|:')
for E in $_PLISTX ; do
if [[ $(printf "$_XREM" | grep -o "${E}") ]] ; then bliner
_BLTFRW="${_SYSLIB}/PrivateFrameworks/BulletinBoard.framework"
_BPLIST="DefaultSectionOrder DefaultEnabledWeeApps"
if [[ -e $_BLTFRW ]] ; then notab echo -e "\n ${_RD}Removal of ${_WH}${E} ${_RD}bundles requires $(allcaps "m")odification
of ${_WH}BulletinBoard ${_RD}plist"
for P in $_BPLIST ; do
function fn() { echo "System/Library/BulletinBoard/$P";}
if grep -iq "${E}" ${_BLTFRW}/${P}.plist 2>/dev/null ; then
plutil -convert xml1 ${_BLTFRW}/${P}.plist > /dev/null
grep -Eiv "${E}" ${_BLTFRW}/${P}.plist > /tmp/${P}.txt && rm ${_BLTFRW}/${P}.plist 2>/dev/null
cat /tmp/${P}.txt > ${_BLTFRW}/${P}.plist && rm /tmp/${P}.txt 2>/dev/null
plutil -convert binary1 ${_BLTFRW}/${P}.plist > /dev/null 2>/dev/null
chmod 664 ${_BLTFRW}/${P}.plist 2>/dev/null
notab echo -ne "\n ${_CY}Edited ${_RS}$(hobble shortd fn) | sed -e "s:/:$(red)\:$(nocolor):g" -e "s:(Default*):$(white)\1:"
; cutxt "${_RS}.plist" ; printf "\n" ; else
notab echo -ne "\n ${_CY}The file ${_RS}$(hobble shortd fn) | sed -e "s:/:$(red)\:$(nocolor):g" -e "s:(Default*):$(white)\1:"
; cutxt "${_RS}.plist" ; printf "${_CY}${_AM}$(allcaps "h")as already been edited\n" ; fi ; done ; fi ; fi ; done
if [[ $(printf "$_XREM" | grep -o "Game") ]] ; then bliner
notab printf "\n ${_RD}Removal of ${_WH}${E} ${_RD}bundles also requires $(allcaps "m")odification of ${_WH}SpringBoard
${_RD}Pref" ; cutxt "erence" ; echo -e "${_RD}s\n"
_PREFSB="${_MOBILE}/Library/Preferences/com.apple.springboard.plist"
function fn() { echo -e "mobile/Library/Preferences/com.apple.springboard" ; }
if grep -iq "Game" $_PREFSB ; then
plutil -convert xml1 $_PREFSB > /dev/null
grep -Eiv "${E}" $_PREFSB > /tmp/springboard.txt && rm $_PREFSB 2>/dev/null
cat /tmp/springboard.txt > $_PREFSB && rm /tmp/springboard.txt 2>/dev/null
plutil -convert binary1 $_PREFSB > /dev/null
chmod 664 $_PREFSB 2>/dev/null
notab echo -ne " ${_CY}Edited ${_WH}$(hobble shortd fn | sed "s:/:\{\$_RD\}\:\{\$_WH\}:g")$(cutxt "${_RS}.plist")\n" ; else
notab printf " ${_CY}The file ${_WH}$(hobble shortd fn | sed "s:/:\{\$_RD\}\:\{\$_WH\}:g")" ; cutxt "${_RS}.plist" ; printf
"${_CY}${_AM}$(allcaps "h")as already been edited\n" ; fi ; fi
#-----
#----- Remove 'StaticBookmarks' from MobileSafari.app ((( except: 'en_US','cs_CZ','fr_FR' ))) -----
#-----
cleanup
_XBOOK=$(grep '^_XLANG' ${LOGS}/rm.logxcept.log 2>/dev/null | tr '|' '\n' | grep '^^[a-zA-Z]\{2\}$' | tr '\n' '|' | sed
's:\([a-zA-Z]\{2\}\):\1:g')
find ${_STAPPS}/MobileSafari.app -type f | grep 'StaticBookmarks' | grep -Eiv "${_XBOOK}en_GB|en_US" >> /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then
bliner ; notab echo -ne "\n ${_BL}Found ${_RS}&${_BL} logged [${_RS}$(icount | sed -e "s:$_BL:\n")
\{\$_WH}StaticBookmarks:" -e '^\<[\>\<1\>\<]\>/s/marks/mark/)\${_BL} in \{\_NL\}\{\_CY}MobileSafari
\${_BL}application\n"
cat /tmp/tf.logrm.log >> /tmp/rm.extra.log ; fi

```

```

#----- Remove unused LinguisticData -----
#
liner ; cleanup ; notab echo -ne "\n ${_RD}Searching for ${_WH}Linguistic Data${_RD}" bundles ; cutxt " to remove${_RS} " ;
etcn
declare -A _LDATA
_LDATA[Cyrl]='Cyrillic' _LDATA[de]='German' _LDATA[es]='Spanish' _LDATA[fr]='French' _LDATA[it]='Italian'
_LDATA[nl]='Dutch' _LDATA[ru]='Russian' _LDATA[zh]='Chinese'
_iDIR="/System/Library/LinguisticData"
_XLANG="$ (grep '^_XLANG' ${_LOGS}/rm.logxcept.log 2>/dev/null | tr '|' '\n' | grep '[a-zA-Z]\{2\}' | tr '\n' '|' | sed
's:\(\)\*: \1en:')"
if [[ $(find "$_iDIR/" -type d 2>/dev/null | grep -E "(echo ${!_LDATA[@]} | sed 's: |:g') | grep -Eww "$_XLANG") ]] ; then
bottom ; notab echo -ne "\n ${_WH}Select ${_CY}Linguistic Data ${_WH} for removal ${_RS}" ; etcnn
for E in $(echo ${!_LDATA[@]} | tr ' ' '\n' | grep -Ev "$_XLANG") ; do
if [[ -e "$_iDIR/$E" ]] ; then blank
bottom ; notab printf "\r ${_BL}Remove ${_RS}${_LDATA[$E]} ${_BL}Linguistic Data${_RS} " ; etc ; yn ; answer ; print_yn
if [[ "$ii" = Y ]] ; then echo "$_iDIR/$E" >> /tmp/tf.logrm.log
blank ; notab echo -ne "\033[1A$(blank) ${_RD}Marking ${_RS}${_LDATA[$E]} ${_RD}for removal" ; sleep 1
else blank ; notab echo -ne "\033[1A$(blank) ${_MG}Keeping ${_RS}${_LDATA[$E]}" ; sleep 1 ; fi ; fi ; done ; fi
if [ -s /tmp/tf.logrm.log ] ; then
notab echo -e "\r ${_RD}Found ${_RS}&${_RD} logged [${_RS}$(icount | sed -e "s: \\\${_RD} \\ \\\${_MG}Linguistic Data
\\\${_RD}files:" -e '/^\<[\>\<1\>\<]\>/s/files/file/')${_RS}\n"
cat /tmp/tf.logrm.log | tee -a /tmp/rm.extra.log | sed -e 's:./,/,|& while read _DATA ; do
echo -e "${_LDATA[_DATA]}\n" | tr '\n' ' ' | sed -e 's: ^: ' ; printf "\n" ; done ;
else blank ; notab echo -ne "\n ${_CY}No ${_WH}Linguistic Data${_CY} bundles to remove \n" ; fi
#----- Remove all images with 'Shadow' in SpringBoard.app -----
#
cleanup
find ${_SYSLIB}/CoreServices/SpringBoard.app -type f -iname *shadow* >> /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then bliner ; notab echo -ne "\n ${_WH}Searching for images with '${_CY}Shadow${_WH}' in the
${_CY}SpringBoard${_WH} app ${_RS}" ; etcn
notab echo -ne "\n Their purpose is cosmetic, but they can slow the SpringBoard\n\n ${_RD}Remove [${_WH}$(icount | sed
-e "s: \\\${_RD} \\ \\\${_WH}Shadow\\\${_RD}" images:" -e '/^\<[\>\<1\>\<]\>/s/ages/age/') found in
${_MG}SpringBoard${_RD}.app ${_RS}" ; etc ; yn ; answer ; print_yn
if [[ "$ii" = Y ]] ; then while read -r _LINE ; do rm -fr "$_LINE" ; done < /tmp/tf.logrm.log
notab echo -ne "\n ${_RD}DONE [${_RS}✓${_RD}]${_CY} $(icount | sed -e "s: \\\${_WH}Shadow\\\${_CY}" images:" -e
'/^\<[\>\<1\>\<]\>/s/ages/age/' -e 's:[0-9]* ::') removed from ${_WH}SpringBoard${_CY}.app\n" ; else
notab echo -ne "\n ${_CY}No ${_BL}'${_WH}Shadow${_BL}' ${_CY}images were removed${_RS} " ; etcn ; fi ; fi
#----- Move FONTS to other partition & free up space on system partition -----
#
if [[ ! -L ${_SYSLIB}/Fonts ]] ; then
cp -R -p ${_SYSLIB}/Fonts ${_PRIVAR}/Fonts
rm -rf ${_SYSLIB}/Fonts
ln -s ${_PRIVAR}/Fonts ${_SYSLIB}
liner ; notab echo -e "\n ${_RD}Folder [${_RS}System${_RD}:${_RS}Library${_RD}:${_RS}Fonts${_RD}] moved to
[${_RS}private${_RD}:${_RS}var${_RD}]\n\n ${_RS}& ${_BL}Symlink created in [${_RS}System${_BL}:${_RS}Library${_BL}]" ; fi
#----- Move AUDIO to other partition & free up space on system partition -----
#
if [[ ! -L ${_SYSLIB}/Audio ]] ; then
cp -R -p ${_SYSLIB}/Audio ${_PRIVAR}/Audio
rm -rf ${_SYSLIB}/Audio
ln -s ${_PRIVAR}/Audio ${_SYSLIB}
notab echo -e "\n ${_RD}Folder [${_RS}System${_RD}:${_RS}Library${_RD}:${_RS}Audio${_RD}] moved to
[${_RS}private${_RD}:${_RS}var${_RD}]\n\n ${_RS}& ${_BL}Symlink created in [${_RS}System${_BL}:${_RS}Library${_BL}]" ; fi
#----- Disable unused LAUNCH DAEMONS - No essential or important daemons are in list -----
#
mkdir -p ${_DOWNL}/Backup/LaunchDaemons 2>/dev/null
chown root:wheel "${_DOWNL}/Backup/LaunchDaemons" 2>/dev/null
cleanup ; liner
notab echo -ne "\n ${_RD}Do you want to ${_WH}Disable ${_RD}or ${_WH}Reenable ${_RD}some ${_NL}Launch Daemons
${_RS}" ; etc ; yn ; answer ; print_yn ; echo
if [[ "$ii" = Y ]] ; then
notab echo -ne " ${_WH}DISABLE${_RS}[${_GR}0${_RS}]${_RS} or ${_WH}REENABLE${_RS}[${_RD}1${_RS}] " ; etc ; ask ; zanswer ;
print_oi

```

```

if [[ "$ff" = 1 ]] ; then
if [[ ! $(find ${_DOWNL}/Backup/LaunchDaemons -type f 2>/dev/null) ]] ; then
notab echo -e "\n ${_RD}There are no disabled Launch Daemons ${_RS}" ; etcn ; else
notab echo -e "\n ${_RS}The following ${_WH}Launch Daemons${_RS} are currently disabled:\n"
find ${_DOWNL}/Backup/LaunchDaemons -type f | sed -e 's,*,/,,' -e 's:\(com\. [a-z]*\.\. \.plist\)::g' | tr [a-z] [A-Z] | cat -n |
awk '{printf "\033[36m%02d\033[0m%s\n", $1, $2}' | tee /tmp/tf.enabledm.log
notab echo -ne "\n ${_RS}Enter ${_CY}number${_RS}(${_CY}${_BK}s${_RS})${_BK} to reenable ${_RS}separated by a
${_WH}single space ${_RS}& hit [${_MG}ENTER${_RS})${_YL}\n\n " ; txt && read _ENABLE
if [[ $_ENABLE = "" ]] ; then notab printf "\033[1A\033[1A\033[1A" ; blank ; notab echo -n " Nothing selected; no daemons
were reenabled" ; else for _DAE in $_ENABLE ; do if [[ ! $(sed -n "${_DAE}p" /tmp/tf.enabledm.log 2>/dev/null) ]] ; then
notab echo -ne "\n$(blank) ${_CY}${_DAE} ${_RS}is not a valid selection number. ${_RD}RETYPE SELECTION${_RS}: ${_CY}"
else
mv -f $(find ${_DOWNL}/Backup/LaunchDaemons -type f | grep -i "$(sed -n "${_DAE}p" /tmp/tf.enabledm.log 2>/dev/null |
sed 's:[^A-Z]*\([A-Z].*\):\1:') ${_SYSLIB}/LaunchDaemons/ 2>/dev/null
notab echo -ne "\n ${_RS}Reenabled: ${_RD}$(sed -n "${_DAE}p" /tmp/tf.enabledm.log 2>/dev/null | sed
's:[^A-Z]*\([A-Z].*\):\1:' | tr [A-Z] [a-z])${_RS}.plist" ; fi ; done ; fi ; fi
notab echo -ne "\n\n ${_WH}Do you also want to ${_RD}Disable${_WH} some Launch Daemons ${_RS}" ; etc ; yn ; answer ;
print_yn ; if [[ "$ii" = Y ]] ; then _DISABLE="YES" ; fi
elif [[ "$ff" = 0 ]] ; then _DISABLE="YES" ; fi ; fi
if [[ "$_DISABLE" = "YES" ]] ; then notab printf "\n ${_RD}Checking for unessential Launch Daemons ${_RS}" ; etcn
for _IDG in i2_ice3 i3_ice3 i4_ice2 ; do
if [[ "$(grep -o '^.[0-9]' ${_LOGS}/rm.settings.log 2>/dev/null)" = ${_IDG%_*} ]] ; then _ICE="$(echo ${_IDG%_*})"
fi ; done
notab echo -e " ${_MG}Essential Daemons ${_RS}are not among the choices\n Any ${_WH}Launch Daemon${_RS} whose
function is ${_GR}unclear\n ${_RS})${_AY}Should be ${_WH}researched *${_RD}before${_WH}*${_RS} it's disabled\n${_MG}
Disabled Daemons${_RS} can of course be re${_GR}enabled"
notab echo -e "\n ${_WH}SELECT${_WH})[${_GR}1${_WH})[${_GR}DISABLE${_RS} ${_NL})[${_AB}Launch Daemon\n
${_WH}SELECT${_WH})[${_RD}0${_WH})[${_RD}TO KEEP${_RS} ${_NL})[${_AB}Launch Daemon\n"
_COUNT=0 ; IFS=$'|'
for E in $_iDAEMONS ; do unset IFS ; bottom
if [[ -e ${_SYSLIB}/LaunchDaemons/${_E} | awk -F '%' '{printf "%s", $1}' | sed -e '/saurik/!s/^/com.apple./' -e
"s:iceX:${_ICE}:" -e 's:/$/plist/' ) ]] ; then let _COUNT++
notab echo -ne " ${_RS})[${_CY})(printf %02d "$_COUNT")${_RS})(echo ${_E^^} | sed -e "s:%:\\${_RS}.plist%:" -e
"s:iceX:"${_ICE^^}:" | awk -F '%' '{ printf "]"\\$_WH%-41.41s\\$_BL$_NL%41s", $1, $2 }') ${_NL})${_RS})[${_GR}1${_RS})[${_RS} or
${_RS})[${_RD}0${_RS}) ${_RS}" ; etc ; ask ; zanswer
case "$ff" in
1) _LDS="DISABLE";;
0) _LDS="KEEPING";;
esac ; notab echo -ne " ${_RS})[${_RD})[${_LDS})${_RS})" ; sleep 0.5 ; blank
if [[ "$ff" = 1 ]] ; then
echo -e "${_E}" | awk -F '%' '{printf "%s", $1}' | sed -e '/saurik/!s/^/com.apple./' -e 's:/$/plist/n/' >> /tmp/rm.daemons.log
fi ; fi ; done ; fi
#-----
# else echo -e "${_E}" | awk -F '%' '{printf "%s", $1}' | sed 's:.*[\\.]\\([A-Za-z]*\\):\\1:' >> /tmp/tf.keepdaemons.log
# cat /tmp/tf.keepdaemons.log 2>/dev/null | sed -e 's:^:_XKEEPDMON|:' -e 's:|$::' ${_LOGS}/rm.logxcept.log
#-----
#----- NOW COPY THEM one by one to backup and remove original -----
#-----
if [ -s /tmp/rm.daemons.log ] ; then
notab echo -e " ${_RD}Disabling the following Launch Daemons${_RS}:\n"
sed 's/com\\. (apple|saurik)\\. //lg' /tmp/rm.daemons.log | tr [a-z] [A-Z] | awk '{printf
"\033[36m%02d\033[37m%-44.44s\n,NR,$1}' | sed -e 's/^/ /' -e 's:(\\.PLIST):$(nocolor)\1:'
notab echo -ne "\n ${_RS}CONFIRM${_BL})[${_GR}Y${_BL})[${_RS} or ${_RS}CANCEL${_BL})[${_RD})N${_BL}) ${_RS}" ; etc ; ask ;
answer ; print_yn
if [[ "$ii" = Y ]] ; then
while read -r _LINE ; do
mv -f ${_SYSLIB}/LaunchDaemons/${_LINE} ${_DOWNL}/Backup/LaunchDaemons/${_LINE} 2>/dev/null ; done <
/tmp/rm.daemons.log
notab echo -ne "\n ${_RD}Disabled Daemons moved to
${_NL})[${_RS}mobile${_RD}):${_RS}Documents${_RD}):${_RS}Backup${_RD}):${_RS}LaunchDaemons${_RD})"
else notab echo -ne "\n ${_RD}CANCELLING${_RS}" ; etc ; printf " ${_RS}Selected Daemons list is in ${_CY}/tmp ${_RS}folder
${_NL}until iDevice reboots\n" ; fi ; else
notab printf " ${_CY}No ${_WH}Launch Daemons${_CY} were disabled ${_RS}" ; etc ; fi
#-----
#----- Modify PLIST com.apple.itunesstored to EXIT* after use -----
#-----
echo ; liner ; notab echo -ne "\n ${_RD}Launch Daemon ${_WH}Modification${_RS}" ; etcn ; notab echo -e "\n
${_RD}Checking whether ${_WH}itunesstored${_RD} & ${_WH}locationd${_RD} ${_NL}Daemons are set to ${_MG})[${_AR}exit

```

```

after use\n"
_iTUNESTR="$_{SYSLIB}/LaunchDaemons/com.apple.itunesstored.plist"
if grep -q "SuccessfulExit" $_iTUNESTR ; then plutil -convert xml1 $_iTUNESTR > /dev/null
tr '\t' '%' < $_iTUNESTR > /tmp/itunesstored.txt && rm $_iTUNESTR
tr '\n' '~' < /tmp/itunesstored.txt > /tmp/itunesstored1.txt && rm /tmp/itunesstored.txt
sed -e
's:KeepAlive</key>~%<dict>~%<key>AfterInitialDemand</key>~%<true/>~%<key>SuccessfulExit</key>~%<false/>
~%</dict>~%:KeepAlive</key>~%<false/>~%:' -e 's:<plist version="1.0">~<dict>:<plist
version="1.0">~<dict>~%<key>EnvironmentVariables</key>~%<dict>~%<key>DYLD_INSERT_LIBRARIES</key>~%<
string>/Library/MobileSubstrate/MobileSubstrate.dylib</string>~%</dict>:' /tmp/itunesstored1.txt >
/tmp/itunesstored.txt && rm /tmp/itunesstored1.txt
tr '~' '\n' < /tmp/itunesstored.txt > /tmp/itunesstored1.txt && rm /tmp/itunesstored.txt
tr '% '\t' < /tmp/itunesstored1.txt > $_iTUNESTR && rm /tmp/itunesstored1.txt
plutil -convert binary1 $_iTUNESTR > /dev/null
notab echo -e " ${_RS}The [$_WH]itunesstored[$_RS] Launch Daemon $_AR}${(allcaps "w")}as modified to exit after use"
else notab echo -e " ${_BL}The [$_RS]itunesstored[$_BL] Launch Daemon $_AM}${(allcaps "h")}as already been modified" ; fi
#-----
#----- Modify PLIST com.apple.locationd.plist to EXIT after use -----
#-----
_LOCATD="$_{SYSLIB}/LaunchDaemons/com.apple.locationd.plist"
if ! grep -q "MobileSubstrate" $_LOCATD ; then plutil -convert xml1 $_LOCATD > /dev/null
tr '\t' '%' < $_LOCATD > /tmp/locationd.txt && rm $_LOCATD
tr '\n' '~' < /tmp/locationd.txt > /tmp/locationd1.txt && rm /tmp/locationd.txt
sed -e
's:KeepAlive</key>~%<dict>~%<key>SuccessfulExit</key>~%<false/>~%</dict>~%:KeepAlive</key>~%<false/>~%:'
-e 's:RunAtLoad</key>~%<true/>:RunAtLoad</key>~%<false/>:' -e 's:<plist version="1.0">~<dict>:<plist
version="1.0">~<dict>~%<key>EnvironmentVariables</key>~%<dict>~%<key>DYLD_INSERT_LIBRARIES</key>~%<
string>/Library/MobileSubstrate/MobileSubstrate.dylib</string>~%</dict>:' < /tmp/locationd1.txt > /tmp/locationd.txt
&& rm /tmp/locationd1.txt
tr '~' '\n' < /tmp/locationd1.txt > /tmp/locationd1.txt && rm /tmp/locationd1.txt
tr '% '\t' < /tmp/locationd1.txt > $_LOCATD && rm /tmp/locationd1.txt
plutil -convert binary1 $_LOCATD > /dev/null
notab echo -e "\n $_RS}The [$_WH]locationd[$_RS] Launch Daemon $_AR}${(allcaps "w")}as modified to exit after use"
else notab echo -e "\n $_BL}The [$_RS]locationd[$_BL] Launch Daemon $_AM}${(allcaps "h")}as already been modified" ; fi
#-----
#----- Modify .GlobalPreferences PLIST to specify custom location -----
#-----
liner ; cleanup ; notab echo -ne "\n $_WH}Set a $_RD}Custom Location[$_WH] by combining two ${(allcaps "d")}ifferent
language codes,$_RS}" ; cutxt "\n"
echo -ne " For example [$_WH}${_AR}en[$_RS]$_AM}_[$_WH]$_AR}FR[$_RS]}" ; cutxt ' '
notab echo -ne "${(allcaps "p")}hysically $_MG}localizes $_RS}the iDevice[$_RS] in $_RS>::$_BK} $_CY}France\n"
notab echo -ne " $_RS}&$_MG}${_AR} Language" ; cutxt "\r" ; echo -ne " Settings $_RS}&$_MG}${_AR} UI Interface
[$_RS]$_AW}for $_NL}${_WH}$_AM}Clock $_RS}$_AW}&$_WH}$_AM} Calendar $_RS}$_AW}&$_WH}$_AM} iDevice" ;
cutxt "$(repeat 8 printf "\b")" ; echo -e " Location[$_RS] ::$_BK} $_CY}English[$_RS]"
_GLOPREF="$_{MOBILE}/Library/Preferences/.GlobalPreferences.plist" ; plutil -convert xml1 $_GLOPREF > /dev/null
function setlocal() { sed -n '/AppleLocale/{n;p;}' $_GLOPREF | grep -Eo "[a-z]{2}[A-Z]{2}" ; }
function setlang() { echo "$_LCODES" | tr '% '\n' | grep -Ew "$ (setlocal | sed 's:.*:::')" | awk -F '|' '{printf "%s", $1}' ; }
function setloc() { echo "$_RCODES" | tr '% '\n' | grep -Eiw "$ (setlocal | sed 's:^\.*:::')" | awk -F '|' '{printf "%s", $2}' | tr '_ ' '!' |
sed 's:.\.: :' ; }
notab echo -ne "\n $_RD}iDevice is currently set to [$_WH}$(setlocal | sed 's:/$(nocolor)_$(white)/')[$_RD]}
i.e. $_CY}$_NL}Language[$_RD]::$_WH}$(setlang)[$_NL}[$_RD]} & $_LS}$_CY}Location[$_RD]::$_WH}$(setloc)[$_RS]}\n"
notab echo -ne "\n $_WH}Select a new $_CY}Custom Locale $_WH}for iDevice $_RS}" ; etc ; yn ; answer ; print_yn ; if [[ "$ii"
= Y ]] ; then notab printf "\033[1A\033[1A" ; blank
notab echo -ne "\n $_WH}Display $_RD}SO 639-1 $_WH}reference list of Language Codes[$_RS]" ; etc ; yn ; answer ;
print_yn ; if [[ "$ii" = Y ]] ; then
printf "\n" ; echo "$_LCODES" | tr '% '\n' | grep -v '~|Emoji|British' | sed
's:^\([A-Z][A-z]{2,}\)*\(\|\)\([A-Z]\{3,\}\)*0*\([a-z][a-z]\)\.?:\4\2\1:g' | awk -F '|' '{printf
"\033[0m[\033[36m%-2s\033[0m] \033[37m%-24.24s\n", $1, $2}' | pa2ste | noindentd sed 's/^/ /' ; fi
notab echo -ne "\n $_YL}Select $_WH}Language $_RS}by code or name & $_YL}press $_RS}[$_CY]ENTER[$_RS]} or
[$_YL}hit $_RS}[$_CY]RETURN[$_RS]} to $_YL}Escape $_RS}:: $_MG}"
txt ; while read _iLANG ; do
if [[ $_iLANG = "" ]] ; then notab echo -e "\033[1A $(repeat 10 printf "\t")\b\b[$_RS][$_RD]CANCELLED[$_RS]" ; break
elif [[ $_iLANG = [^[:alpha:]]* || ! $(echo "$_LCODES" | tr '% '\n' | grep -v '~' | grep -Eiw "$_iLANG") ]] ; then
notab printf "\033[1A $_RS}$_4LINE}$_YL}INVALID[$_RD]}[$_WH}SELECTION[$_RS]$_4LINE}" ; sleep 1
notab printf "\r $_RS}$_4LINE}$_YL}PLEASE[$_RD]}SELECT[$_WH}AGAIN[$_RS]$_4LINE}" ; sleep 0.5
notab printf "\r $_YL}Select $_WH}Language $_RS}by code or name & $_YL}press $_RS}[$_CY]ENTER[$_RS]} or $_YL}hit
[$_RS}[$_CY]RETURN[$_RS]} to $_YL}Escape $_RS}:: $_MG}" ; else
_XLOCALANG=$(echo "$_LCODES" | tr '% '\n' | grep -v '~' | grep -Eiw "$_iLANG" | sed 's:^\.*\([a-z]\{2\}\)\.?:\1:g') ; break

```

```

; fi
notab echo -ne "\n ${_WH}Display ${_RD}ISO 3166-1-α2 ${_WH}reference list of Country Codes${_RS} " ; etc ; yn ; answer ;
print_yn ; if [[ "$_i" = Y ]] ; then
printf "\n" ; echo "$_RCODES" | tr '% '\n' | awk -F '|' '{printf "\033[0m[\033[36m%-2s\033[0m
\033[37m%-24.24s\n", $1, $2}' | pa2ste | noindentd sed 's/^/ /' | sed -e 's: :g' -e 's:&:(nocolor)&(white):g' -e
"s:-:(nocolor)-$(white):g" ; fi
notab echo -ne "\n ${_YL}Select ${_WH}Location ${_RS}by code or name ${_RS}&${_YL} press ${_RS}[${_CY}ENTER${_RS}] ::
${_MG}"
txt && while read _iLOC ; do
if [[ $_iLOC = "" || $_iLOC = [^[:alpha:]]* || ! $(echo "$_RCODES" | tr '% '\n' | grep -Ei "$(echo $_iLOC | tr ' ' '_')") ]] ; then
notab printf "\033[1A ${_RS}${_4LINE}${_YL}INVALID${_RD}†${_WH}SELECTION${_RS}${_4LINE}" ; sleep 1
notab printf "\r ${_RS}${_4LINE}${_YL}PLEASE${_RD}SELECT${_WH}AGAIN${_RS}${_4LINE}" ; sleep 0.5
notab printf "\r ${_YL}Select ${_WH}Location ${_RS}by code or name ${_RS}&${_WH} press ${_RS}[${_CY}ENTER${_RS}] ::
${_MG}" ; else
_XLOCALANG="$(echo "$_RCODES" | tr '% '\n' | grep -Ei "$(echo $_iLOC | tr ' ' '_')" | awk -F '|' '{printf "%s", $1}' | sed
"s:^:$_XLOCALANG:")" ; break ; fi ; done
_XEDIT="$(grep '^_XLANG' ${_LOGS}/rm.logxcept.log 2>/dev/null | sed 's:_XLANG::' | tr '|' '\n' | grep -E '^0*[a-zA-Z]{2}0*$'
| sed -e 's:^\([a-z]{2}\)$:<string>\1</string>:g' -e 's:\(0[a-z]{2}\)$:\1</string>:g' -e 's:\(^[a-z]{2}0\):<string>\1:g'
-e 's:0:-:g)"
cat ${_GLOPREF} | sed -e 's:^\ [t]*<string>[a-z]{2}\(-[A-z]{2,}\)*</string>::g' | sed -n '1,/^\$/p' | grep '.' >
/tmp/tf.edit.log
grep -E "${_XEDIT}>en<" ${_GLOPREF} >> /tmp/tf.edit.log
cat ${_GLOPREF} | sed -e 's:^\ [t]*<string>[a-z]{2}\(-[A-z]{2,}\)*</string>::g' | sed -n '/^\$/,$p' | grep '.' | sed
"s:[a-z][a-z]_[A-Z][^<]*:$_XLOCALANG:" >> /tmp/tf.edit.log
cat /tmp/tf.edit.log > ${_GLOPREF}
notab echo -ne "\n ${_RD}iDevice has been set to [${_WH}$(setlocal | sed "s/_/$(nocolor)_$(white)/")${_RD}]
${_CY}${_NL}Language${_RD}::${_WH}$(setlang)${_NL}${_RD} & ${_LS}${_CY}Location${_RD}::${_WH}$(setloc)${_RS}\n" ; done
else notab echo -ne "\033[1A$(blank) ${_LS}${_BL}iDevice ${_RS}Localization Settings${_BL}" ; cutxt " were" ; printf "${_BL}
not modified ${_RS}" ; etcn ; fi
plutil -convert binary1 ${_GLOPREF} > /dev/null
#-----
#----- FIRST RUN CREATE USEFUL SYMLINKS -----
#-----
#----- Symlink ((( Documents ))) to ((( mobile:Media:Downloads ))) in useful places -----
#-----
if [[ ! -L ${_MOBILE}/Documents || ! -L /Documents ]] ; then bliner
ln -s ${_DOWNL} ${_MOBILE}/Documents 2>/dev/null
ln -s ${_DOWNL} /Documents 2>/dev/null
chown -h mobile:mobile /mobile/Documents
notab echo -e "\n ${_RD}Symlink [${_RS}Documents${_RD}] to directory
[${_RS}mobile${_RD}]:[${_RS}Media${_RD}]:[${_RS}Downloads${_RD}] created in [${_RS}mobile${_RD}] ${_RS}&${_RD}
[${_RS}/${_RD}]"
fi
#-----
#----- Symlink to ((( mobile ))) in useful places -----
#-----
if [[ ! -L /mobile ]] ; then bliner
ln -s ${_MOBILE} / 2>/dev/null
ln -s ${_MOBILE} ${_PRIVAR}/stash 2>/dev/null
ln -s ${_MOBILE} ${_PRIVAR}/root 2>/dev/null
bliner ; notab echo -e "\n ${_RD}Symlink [${_RS}mobile${_RD}] created in directories
[${_RS}private${_RD}]:[${_RS}var${_RD}]:[${_RS}root${_RD}] ${_RS}&${_RD}
[${_RS}private${_RD}]:[${_RS}var${_RD}]:[${_RS}stash${_RD}] ${_RS}&${_RD} [${_RS}/${_RD}]"
fi
#-----
#----- Symlink to ((( Preferences ))) in useful places -----
#-----
if [[ ! -L ${_MOBILE}/.Preferences ]] ; then bliner
ln -s ${_MOBILE}/Library/Preferences ${_MOBILE}/.Preferences 2>/dev/null
notab echo -e "\n ${_RD}Symlink [${_RS}.Preferences${_RD}] to directory
[${_RS}mobile${_RD}]:[${_RS}Library${_RD}]:[${_RS}Preferences${_RD}] created in [${_RS}mobile${_RD}]"
fi
#-----
#----- Create symlink to ((( Applications ))) in useful places -----
#-----
if [[ ! -L ${_MOBILE}/.iApplications ]] ; then
ln -s ${_STAPPS} ${_MOBILE}/.iApplications 2>/dev/null
notab echo -e "\n ${_RD}Symlink [${_RS}.iApplications${_RD}] to directory

```

```

[{$_RS}private{$_RD}:{$_RS}var{$_RD}:{$_RS}stash{$_RD}:{$_RS}Applications{$_RD}] created in [{$_RS}mobile{$_RD}]"
fi
#-----
#----- Create symlink to ((( Themes* ))) directory in useful places -----
#-----
_WINTER="$(find $_PRIVAR}/stash/Themes.* -prune 2>/dev/null)"
if [[ -d $_WINTER ]] ; then
if [[ ! -L $_MOBILE}/Themes ]] ; then bliner
ln -s $_WINTER $_MOBILE}/Themes
ln -s $_WINTER /Themes 2>/dev/null
notab echo -e "\n $_RD}Symlink [$_RS}Themes{$_RD}] created in [$_RS}/$_RD] & [$_RS}mobile{$_RD}]"
fi ; fi
#-----
#----- Create symlink to ((( TimeCapsule ))) in useful places -----
#-----
if [[ ! -L $_DOWNL}/TimeCapsule ]] ; then bliner
if [[ ! -d $_MOBILE}/Library/Preferences/TimeCapsule ]] ; then
mkdir -p $_MOBILE}/Library/Preferences/TimeCapsule
chown -R mobile:mobile $_MOBILE}/Library/Preferences/TimeCapsule ; fi
ln -s $_MOBILE}/Library/Preferences/TimeCapsule $_DOWNL} 2>/dev/null
notab echo -e "\n $_RD}Symlink [$_RS}TimeCapsule{$_RD}] created in [$_RS}mobile{$_RD}:$_RS}Documents{$_RD}]"
fi
#-----
#----- Create symlink ((( Attachments ))) linked to ((( Downloads ))) -----
#-----
_ATTMT="$_MOBILE}/Library/Attachments"
if [[ ! -L $_ATTMT ]] ; then
if [[ -d $_ATTMT ]] ; then
mv -f $_ATTMT/* $_DOWNL}/ 2>/dev/null
rmdir $_ATTMT ; fi ; bliner
ln -s $_DOWNL} $_MOBILE}/Library/Attachments
chown -h mobile:mobile $_ATTMT}
notab echo -e "\n $_RD}Mobile directory [$_RS}Media{$_RD}:$_RS}Downloads{$_RD}] symlinked to
[$_RS}Attachments{$_RD}] in [$_RS}mobile{$_RD}:$_RS}Library{$_RD}]"
fi
#-----
#----- Create other useful symlinks in useful places -----
#-----
if [[ ! -d /deb ]] ; then bliner
mkdir -p /deb
notab echo -e "\n $_RD}Root directory [$_RS}deb{$_RD}] created" ; fi
if [[ ! -L $_MOBILE}/BackToDeb ]] ; then bliner
mv -f $_MOBILE}/BackToDeb/* /deb/ 2>/dev/null
rmdir $_MOBILE}/BackToDeb
ln -s /deb $_MOBILE}/BackToDeb 2>/dev/null
chown -h mobile:mobile $_MOBILE}/BackToDeb
notab echo -e "\n $_RD}Root directory [$_RS}deb{$_RD}] symlinked to [$_RS}BackToDeb{$_RD}] in [$_RS}mobile{$_RD}]" ;
fi
#-----
iPDF="$(find $_MOBAPP}/*/* -type d -prune | grep -i 'AirSharing' | sed -e 's,[^/]*$,,)"
if [[ -d $iPDF ]] ; then
if [[ ! -L $_DOWNL}/iPDF ]] ; then bliner
ln -s $iPDF/Documents $_DOWNL}/iPDF
chown -h mobile:mobile $_DOWNL}/iPDF
notab echo -e "\n $_RD}Symlink [$_RS}iPDF{$_RD}] created in [$_RS}mobile{$_RD}:$_RS}Documents{$_RD}]" ; fi ; fi
#-----
##### SYMLINKS #####
#-----
#----- First remove dead symlinks -----
#-----
cleanup ; liner ; notab echo -ne "\n $_RD}Searching for $_WH}dead symlinks $_RD}to remove $_RS}" ; etc
find $Path -type l ! -xtype f ! -xtype d | tee -a /tmp/tf.logrm.log | xargs -0 -l -d '\n' rm -f
if [[ -s /tmp/tf.logrm.log ]] ; then notab echo -e "\n\n $_WH}Found [$_RD]$(icount | sed -e "s:$(white)} dead symlinks:"
-e '/^\<[\>\<1\>\<]\>/s/links/link/)"
else notab echo -e "\n\n $_CY}$_AM}Found no $_WH}dead symlinks $_CY}on iDevice{$_RS}" ; fi
#-----
if [[ ! -L /AppLinks ]] ; then bliner
if [[ ! -d $_MOBILE}/AppLinks ]] ; then

```



```

mkdir -p ${_MOBILE}/AppLinks
chown mobile:mobile ${_MOBILE}/AppLinks ; fi
ln -s ${_MOBILE}/AppLinks /
ln -s ${_MOBILE}/AppLinks ${_PRIVAR}/stash
notab echo -e "\n ${_RD}Symlink [${_RS}AppLinks${_RD}] created in [${_RS}private${_RD}:${_RS}var${_RD}:${_RS}stash${_RD}]
& [${_RS}/${_RD}]" ; fi
#-----
#----- Create Symlinks to MobileApps in ((( AppLinks ))) directory -----
#-----
cleanup ; bliner ; notab echo -ne "\n ${_CY}Creating ${_WH}Mobile App ${_CY}symlinks ${_RS}${_NL}" ; etcnn
find "${_MOBILE}/AppLinks" -maxdepth 1 -type l | xargs -0 -l -d '\n' rm -f
IFS=$'\n'
find $_MOBAPP -maxdepth 2 -name '*.app' -type d | sed -e "s/\^//g" -e "s/\^//g" |& while read _LINK ; do
_iCODE=$(dirname "$_LINK")
_COUNT=0
if grep -q "^$(basename "$_LINK")$" /tmp/tf.logrm.log 2>/dev/null ; then
let _COUNT++ ; fi
_iAPPL=$(basename "$_LINK" | sed -e "s/(\.app)/\-$_COUNT}/1/" -e 's:\-0::' | tee -a /tmp/tf.logrm.log)
ln -s $_iCODE ${_MOBILE}/AppLinks/$_iAPPL
done ; unset IFS
sort -f /tmp/tf.logrm.log | sed -e 's/^[^_A-z0-9.\&\'"]//g' -e 's/\.app$//' -e 's/(.{15})\.[A-z](.)/\1^\2/' -e 's:[
]::g' | awk -F '%' '{printf "\033[34m%003d\033[0m%-18.18s\n",NR,$0}' | pa3ste | noindentd sed 's/^\^/' | grep '!'
notab echo -e "${_BL}\n Created [${_RS}$(icount)${_BL}] MobileApp symlinks in $(allcaps 'D')irectory
${_RS}private${_BL}:${_RS}var${_BL}:${_RS}mobile${_BL}:${_RS}AppLinks${_BL}"
chown -R mobile:mobile ${_MOBILE}/AppLinks
#-----
#----- Find, Remove wrongly created AppLink symlinks from command above -----
#-----
# find /private/var/mobile/Applications -maxdepth 2 -type l | grep '.*\-....\-....\-....\-.....$' | xargs -0 -l -d '\n' rm -f
#-----
#----- Remove all MobileApp Symlink-Prefs - for testing -----
#-----
# find /private/var/mobile/Applications -type l | grep 'Library/Preferences' | xargs -0 -l -d '\n' rm -f
#-----
#----- Restore missing symlinked PLISTs in MobileApps Library/Prefs folder -----
#-----
_iPLIST='com.apple.PeoplePicker.plist .GlobalPreferences.plist'
cleanup ; for _iSP in $_iPLIST ; do
find $_MOBAPP -mindepth 1 -maxdepth 1 -type d | grep '.*\-.*\-.*\-.*\-' |&
while read _APP ; do if [[ ! -e "${_APP}/Library/Preferences/${_iSP}" ]] ; then
ln -s ${_MOBILE}/Library/Preferences/${_iSP} ${_APP}/Library/Preferences 2>/dev/null
chown -R mobile:mobile ${_APP}/Library/Preferences
basename "$(find $_APP -name *.app -type d -prune)" >> /tmp/tf/${_iSP}/com.apple/}.log
fi ; done ; if [[ -s /tmp/tf/${_iSP}/com.apple/}.log ]] ; then bliner
notab echo -e "\n ${_WH}${_AB}Missing Pref-Symlink ${_NL}${_CY}${_iSP} ${_WH}${_NL}${_AB}restored to [${_RD}$(wc -l <
/tmp/tf/${_iSP}/com.apple/}.log | sed -e "s:$(white) Mobile Apps:" -e '/^\<[\>\<1\>\<1\>\<1\>/s/Apps/App/')\n"
sort -f /tmp/tf/${_iSP}/com.apple/}.log | sed -e 's:\.app::' -e 's/(.{12})\.[A-z](.)/\1^\2/' | awk -F '\n' '{printf
"\033[31m%03d\033[0m%-18s\n",NR,$1}' | pa3ste | noindentd sed 's/^\^/' | grep '!' ; fi ; done
#-----
#-----
fi
#-----
##### ICONOGRAPHY ::: ICON SYNC #####
#-----
if [[ "$(grep 'ICONOGRAPHY' ${_LOGS}/rm.settings.log | awk -F '_' '{printf $1}' )" == "YES" ]] ; then
#-----
#-----
_SYNC="YES"
liner ; notab echo -ne "\n ${_RS}READ ${_WH}INSTRUCTIONS ${_RS}for manual ${_WH}iDevice Theming ${_RS}." ; etc ; yn
answer ; print_yn
if [[ "$_i" = Y ]] ; then notab echo -ne "\033[1A\033[1A$_NAVOD]" ; read -p "" ; else printf "\n" ; fi
while [[ ! -e "${_DOWNL}/Iconography" ]] ; do
notab echo -e "\n ${_WH}Cannot find the icon source directory${_RS}
'/private/var/mobile/Media/Downloads/${_CY}Iconography${_RS}'"
notab echo -ne "\n ${_RS}[${_GR}O${_RS}][${_GR}PAUSE${_RS}]:${_WH}LOCATE ${_RS}&${_WH} UPLOAD ${_RS}your Icon Source
directory, ${_BL}or\n ${_RS}[${_RD}1${_RS}][${_RD}ABORT${_RS}]:${_WH}CANCEL ${_RS}&${_WH} FINISH ${_RS}this another time
..." ; etc ; oi ; zanswer
case "$ff" in

```

```
0) _T=" ${_RS}${_GR}PAUSE${_RS}";
1) _T=" ${_RS}${_RD}ABORT${_RS}";
esac ; printf "$_T" ; if [[ "$ff" = 1 ]] ; then unset _SYNC ; break ; fi
notab echo -ne "\n$(liner)$(liner)\r\t\t\t\t\tHIT${_RD}-${_WH}ENTER${_RD}-${_RS}TO${_RD}-${_RS}CONTINUES${_RD}\n"
read -p "" ; done
while [[ $_SYNC = "YES" ]] ; do cleanup
declare -A _DEST
_DEST[Stash]='/private/var/stash/'
_DEST[Mobile]='/private/var/mobile/Applications/'
_DEST[System_/PreferenceBundles]='/System/Library/PreferenceBundles/'
_DEST[System_/CoreServices]='/System/Library/CoreServices/'
_DEST[System_/Internet Plug-Ins]='/System/Library/Internet Plug-Ins/'
_DEST[System_/Frameworks]='/System/Library/Frameworks/'
_DEST[System_/PrivateFrameworks]='/System/Library/PrivateFrameworks/'
_DEST[System_/WeeAppPlugins]='/System/Library/WeeAppPlugins/'
_DEST[Library_/PreferenceBundles]='/Library/PreferenceBundles/'
_DEST[Library_/PreferenceLoader/Preferences]='/Library/PreferenceLoader/Preferences/'
_DEST[Library_/Activator]='/Library/Activator/'
_DEST[Library_/ActionMenu]='/Library/ActionMenu/'
_DEST[Library_]='/Library/'
_DEST[Audio]='/private/var/Audio/'
_ICONS=''/private/var/mobile/Media/Downloads/Iconography/'
notab echo -ne " ${_WH}READING${_RS} Icon Source directory tree "
for E in "${_DEST[@]}"; do
find "$_ICONS/$E/" -mindepth 1 -maxdepth 1 -type d | grep -v 'Extra_' | sed -e "s:^\.$E\|/:" -e "s:$/:" | sort -fu |&
while read _PATH ; do IFS=${_n}
find "$_DEST[$E]" -mindepth 1 -maxdepth 2 \( -name "${_PATH%/*}*" \) -type d 2>/dev/null | tee -a /tmp/tf.dest.log |
sed "s:${_DEST[$E]}.*:${_ICONS}/${E}/${_PATH%}/:" | sort -f >> /tmp/tf.src.log ; unset IFS ; done ; done
sed 's/ /+/g' /tmp/tf.src.log | tr ' ' '% ' | awk -F '%' '{print $NF}' | tr ' ' ' ' > /tmp/tf.icon_src.log
sed 's/ /+/g' /tmp/tf.dest.log | tr ' ' '% ' | awk -F '%' '{print $NF}' > /tmp/tf.icon_dest.log
notab echo -ne "\n\n${_RD}Check Sync List${_RS} before ${_MG}proceeding ${_RS}${_NL}" ; repeat 15 printf ' '
yn ; answer ; print_yn
if [[ "$ii" = Y ]] ; then
notab echo -ne "$(liner)\r ${_YL}SOURCE${_RD}$(repeat 17 printf "-")${_CY}DESTINATION\n" && sleep 1
paste /tmp/tf.icon_src.log /tmp/tf.icon_dest.log | sed -e
's:\([A-z0-9]\{4,\}\)\(Preferences\|Prefs\|Settings\|Toggle\)\|1/Ig' -e 's:\([A-z]\{2,\}\.\[A-z0-9\]\{2,\}\.):\g' -e
's:\([A-z0-9]\{2,\}\+[A-z0-9]\{2,\}\)\(\+[A-z0-9]\{2,\}\)*\+[A-z0-9]\{2,\}\:1:' -e 's:\.\[A-z0-9\]\{2,\}\:g' | awk '{printf
"\n+\033[33m%-27s\033[36m%-27s", $1"\033[0m", $2}' | sed -e 's: :g' -e 's/+/ /g' -e 's:\([A-z0-9]\)\. \. *$:1:g'
echo -ne "\n\n${_YL}SOURCES ${_RS}...${_RD} MATCH ${_RS}....${_CY} DESTINATIONS ${_RS}....${_BK}${_BK}..
${_WH}VERIFY${_NL}" ; yn ; answer ; print_yn
if [[ "$ii" = N ]] ; then
notab echo -ne "\n${_WH}SYNC CANCELLED${_RS}; There's obviously a problem.. ${_NL}${_RD}Please contact
${_CY}MANAGEMENT\n" ; unset _SYNC ; fi ; fi
if [[ "$_SYNC" == "YES" ]] ; then IFS=${_n}
notab echo -e "$(liner)\r\t\t\t\t\tb\b\b${_LS}${_WH}SYNCHRONIZING\n"
while read -r _SYNCFROM && read -r _SYNCTO <&3 ; do
#-----
#---- Simulate RSYNC command by adding ((( n ))) to options as in ((( rsync -naitOvvv ))) before running it - add ((( I )))
#---- That is ((( capital-letter-i ))) to sync without regard for modification dates - i.e. sync all existing not just newest
#-----
rsync -aitOvvv --stats --existing $_SYNCFROM/ $_SYNCTO 2>/dev/null | sed -n 's/\(get_local_name count\)=.*\|Number
of files transferred\| \|)/%/p' | paste - - | grep -v '%0' | sort | tee -a /tmp/tf.tmp.icons.log | awk -F '%' '{printf "
\033[31m%4d\033[0m UI Images in %s\n", $NF, "\033[36m"$2"\033[0m"}' | cutend | sed -e
'/\<[\>\<1\>\<]\>/s/mages/mage /'
done < /tmp/tf.src.log 3</tmp/tf.dest.log ; unset IFS
if [[ ! -s /tmp/tf.tmp.icons.log ]] ; then
notab echo -ne "\r ${_CY}UPDATE UNNECESSARY${_RS} THERE ARE ${_WH}NO NEW ICONS${_RS} TO SYNC " ; etcn
else awk -F '%' '{printf "%4d > %s\n", $NF, $2}' /tmp/tf.tmp.icons.log > /tmp/tf.icons.log
fi ; fi ; break ; done
#-----
#-----
fi
#-----
##### REMOVE EXTRA #####
#-----
if [[ "$(grep 'RMEEXTRA' $_LOGS}/rm.settings.log | awk -F ' ' '{printf $1}'" == "YES" ]] ; then
#-----
```

```

#-----
#----- Language pack removal - removes all except ENGLISH - Customize on-the-fly to add exceptions -----
#-----
liners ; cleanup
notab echo -ne "\n ${_RD}Remove unused ${_WH}Language Packs ${_RS}" ; etc ; yn ; answer ; print_yn
if [[ "$ii" = Y ]] ; then
notab echo -e "\033[1A ${_CY}Language Packs ${_RS}allow the user to ${_WH}interface ${_RS}with ${_MG}System${_RS} &
${_MG}Applications${_RS}\n In the ${_WH}language${_RS} of their ${_WH}choice${_RS}; usually only 1 language is ever
used${_WH}\n Max 2 ${_RS}if the iDevice's language is set to ${_WH}anything${_RS} other than ${_WH}English${_RS}\n Since
${_WH}English${_RS} is still ${_WH}required${_RS} by the ${_WH}system ${_RS}& ${_RD}must never be removed\n${_RS} This
means there can be ${_WH}thousands${_RS} of unused langpacks in your iDevice\n ${_MG}Dead weight${_RS} which
${_WH}slows ${_RS}down system processes ${_RS}& only ${_WH}fills ${_RS}disk ${_WH}space\n"
if grep -q '^_XLPACK' ${_LOGS}/rm.logxcept.log 2>/dev/null ; then
notab echo -ne " ${_CY}A ${_WH}Langpack exceptions ${_CY}list was found on yr iDevice ${_RS}" ; etcn
notab echo -e "\n ${_RS}Preserving all [${_WH}$(grep '^_XLPACK' ${_LOGS}/rm.logxcept.log | sed -e 's:_XLPACK|::' | tr '|' '\n'
| grep '[A-Z]' | tr '\n' '&' | sed -e 's:\&:${(nocolor)}\:' -e 's:\&:${(nocolor)}\ \& \[($white):g')${_RS} as well as
[${_WH}English${_RS}] language packs\n" ; fi
_XLPACK="$(grep '^_XLPACK' ${_LOGS}/rm.logxcept.log 2>/dev/null | sed -e 's:_XLPACK|::' -e 's:\([a-z]\)\(\|\):1\.\lproj\2:g'
-e 's:0:[-]:g)"
notab echo -ne " ${_RD}Searching for ${_WH}Language Packs${_RD} on entire iDevice ${_RS}" ; etcnn
find $Path -name *.lproj -type d | grep -Eiv "English|en|.lproj|en.US" | grep -Ewv "${_XLPACK}\.lproj" | tee
/tmp/tf.logrm.log | xargs -0 -d '\n' du -cbh | tail -1 > /tmp/tf.size.log
cat /tmp/tf.logrm.log | sed -e 's:.,*,,' -e 's:\.lproj/' | sort -fu > /tmp/tf.lcodes.log
echo $_LCODES | sed 's:%:\n:g' | grep -Eiw "$(cat /tmp/tf.lcodes.log | sed -e
's:\(en|fr|de|es\)\([-_]\)\([a-zA-Z0-9]*\):\2\3:g' -e 's:\([a-zA-Z]\{2}\[-_]\).*:1:g' -e 's:[-]:0:g' -e 's:^:\<' -e 's:$:\>:'
| uniq) | awk -F '|' '{printf "%s\n", $1}' > /tmp/tf.lang.log
if [ -s /tmp/tf.logrm.log ] ; then
function itemize() { "$@" | awk '{printf "\033[36m%02d\033[0m%-10.10s\n",NR,$1}' | paste -- -- | sed 's:^: :'}
itemize cat /tmp/tf.lang.log
if ! grep -q '^_XLPACK' ${_LOGS}/rm.logxcept.log 2>/dev/null ; then _TLOG="lang"
editlist ; _PRINT=$(itemize cat /tmp/tf.lang.log) ; editprint ; _XCPT="LPACK" ; logxlang
grep -Ei "${(echo $_LCODES | sed -e 's:%:\n:g' | grep -Ei "$(cat /tmp/tf.lang.log 2>/dev/null)")} /tmp/tf.lcodes.log >
/tmp/tf.lcodes2.log ; cat /tmp/tf.lcodes2.log > /tmp/tf.lcodes.log ; fi
_EDIT=$(cat /tmp/tf.lcodes.log | tr '\n' '|' | sed 's:|::')
grep -Ei "${_EDIT}\.lproj$" /tmp/tf.logrm.log | tee /tmp/rm.lproj.log | xargs -0 -d '\n' du -cbh | tail -1 > /tmp/tf.size.log
notab echo -e "\n ${_RD}Actual languages to remove [${_WH}$(wc -l < /tmp/tf.lang.log)${_RD}]"
notab echo -e "\n ${_MG}Unique language codes [${_WH}$(wc -l < /tmp/tf.lcodes.log)${_MG}]"
notab echo -e "\n ${_CY}Total ${_WH}Language Packs${_CY} to remove from iDevice [${_WH}$(icount)${_CY}] for
[${_WH}$(isize)${_CY}]"
else notab echo -ne " ${_RS}Found no ${_WH}Language Packs${_RS} to remove " ; etcn ; fi
#-----
#----- Remove ID.Bundle unused language plists (SMS Delivery) -----
#-----
_XID="$(grep '^_XLPACK' ${_LOGS}/rm.logxcept.log 2>/dev/null | tr '|' '\n' | grep -Ev '[A-Z][a-z]{3}' | sed -e 's:0:[-]:g' -e
's:\([a-z]\)\$1\.\plist:' | tr '\n' '|')"
find "/Library/Application Support/ID.bundle" -iname "*.plist" 2>/dev/null | grep -Ev "${_XID}en\.\plist" | grep -v
'deliveryreports' > /tmp/tf.logid.log
if [ -s /tmp/tf.logid.log ] ; then bliner ; cat /tmp/tf.logid.log >> /tmp/rm.extra.log
notab echo -ne "\n ${_RD}Found ${_RS}&${_RD} Logged [${_RS}$(wc -l < /tmp/tf.logid.log | sed -e 's:$(red)\$
(nocolor)language plists:" -e '/^\<[\>\<1\>\<]\>/s/plists/plist/')${_RD} in
${_MG}Library${_RD}:${_MG}AppSupport${_RD}:${_MG}ID.bundle\n" ; fi
#-----
#----- Remove 3 Japanese dictionaries if not used -----
#-----
if grep -q 'Japanese' /tmp/tf.lang.log ; then
find /Library/Dictionaries/ -type d -name 'Shogakukan*' 2>/dev/null > /tmp/tf.logdict.log
if [ -s /tmp/tf.logdict.log ] ; then cat /tmp/tf.logdict.log >> /tmp/rm.extra.log ; bliner
notab echo -ne "\n ${_WH}Found ${_RS}&${_WH} logged [${_RD}$(wc -l < /tmp/tf.logdict.log | sed -e 's:$(white)\$
(nocolor)Japanese$(white) dictionaries:" -e '/^\<[\>\<1\>\<]\>/s/aries/ary/') in
${_MG}Library${_RD}:${_MG}Dictionaries\n" ; fi ; fi ; fi
#-----
#----- Look for Miscellaneous junk or skip ahead? -----
#-----
liners ; notab echo -ne "\n ${_RD}Remove ${_WH}miscellaneous junk ${_RS}" ; etc ; yn ; answer ; print_yn ; if [[ "$ii" = Y ]] ;
then
#-----
#----- Remove *fw[3456]* files not needed by your firmware -----
#-----

```

```

cleanup ; if [[ ! -e ${_LOGS}/rm.settings.log ]] ; then bliner
notab echo -ne "\n ${_WH}Select iDevice ${_CY}Firmware${_RS}: ${_MG}iOS version ${_NL}${_WH}${_YL}3${_WH}${_RS} or
${_WH}${_CY}4${_WH}${_RS} or ${_WH}${_MG}5${_WH} ${_RS} or ${_WH}${_RD}6${_WH} ${_RS}" ; etc ; ask ; while read -s -n
1 fw ; do
echo -e " ${_RD}${_RS}FW${_WH}${fw}${_RD}${_RS}"
if [[ "$fw" = 3 || "$fw" = 4 || "$fw" = 5 || "$fw" = 6 ]] ; then echo FW$fw >> ${_LOGS}/rm.settings.log ; break ; fi ; erata ;
done ; fi
_FW="$(grep -o 'FW[0-9]' ${_LOGS}/rm.settings.log | uniq)"
find $Path -iname '*fw[0-9]*' | grep -iv "${_FW}" > /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then bliner
notab echo -e "\n ${_WH}Found ${_RS}&${_WH} logged [${_RD}$(icount)${_WH}] unused ${_CY}$(echo 'FW2 FW3 FW4 FW5
FW6' | sed -e "s: ${_FW}::" -e "s: : $(nocolor)- $(cyan):g") ${_WH}other${_RS}-${_WH}firmware${_RS}-${_WH}specific files" | sed
'/[^\0-9]1[^\0-9]/s/files/file/' ; cat /tmp/tf.logrm.log >> /tmp/rm.extra.log ; fi
#-----
#----- Remove all Mobile.app 'Library.bak' & 'Documents.bak' created when restoring app-data with Chronus -----
#-----
cleanup ; find ${_MOBAPP} \ ( -name 'Library.bak' -o -name 'Documents.bak' \ ) > /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then bliner
notab echo -e "\n ${_WH}Removed [${_RD}$(icount)${_WH}] empty mobile${_RS}-${_WH}app data folders
${_CY}Document${_RS}.bak${_WH} ${_WH}& ${_CY}Library${_RS}.bak${_WH}\n ${_RS}Orphaned after restoring personal
${_CY}mobile-app-data${_RS} with backup app ${_MG}Chronus" ; cat /tmp/tf.logrm.log >> /tmp/rm.extra.log ; fi
#-----
#----- Check if SetupAssistant LaunchDaemon is disabled, in which case remove Setup.app & Bundles -----
#-----
if [[ ! -e "${_SYSLIB}/LaunchDaemons/com.apple.SpringBoard.SetupHelper.plist" ]] ; then
rm -fr ${_STAPPS}/Setup.app ; rm -fr ${_SYSLIB}/SetupAssistantBundles ; fi
#-----
#----- Find & remove AppStore app 'iTuneMetada.plist' files ((( installed w/Apps to track App use ))) -----
#-----
bliner ; cleanup ; notab echo -ne "\n ${_RD}Search for ${_WH}iTunesMetadadata${_RD} property lists ${_RS}" ; etc ; yn ; answer ;
print_yn
if [[ "$ii" = Y ]] ; then
notab printf "\033[1A ${_WH}Apple Store ${_CY}tracks applications-use ${_RS}via the ${_WH}iTuneMetadadata ${_CY}plist
${_RS}\n Removing Metadata stops app-tracking without limiting functions\n ${_MG}Except
${_WH}in${_RS}-${_WH}App${_RS}-${_WH}Purchase ${_RS}hacks: ${_MG}some ${_WH}do require${_RS} the metadata info\n"
if grep -q '^_XMETA' ${_LOGS}/rm.logxcept.log 2>/dev/null ; then
notab echo -ne "\n ${_CY}An iTunesMetadadata ${_WH}exceptions list${_CY} was found on yr iDevice ${_RS}::\n\n"
notab echo -ne " ${_RS}[${_GR}0${_RS}]${_WH}RE${_RS}-${_WH}USE EXCEPTIONS LIST ${_BL}or\n
${_RS}[${_RD}1${_RS}]${_WH}DELETE ${_RS}&${_WH} MAKE NEW LIST ${_RS}....." ; etc ; oi ; zanswer ; print_oi
if [[ "$ff" = 1 ]] ; then grep -v '^_XMETA' ${_LOGS}/rm.logxcept.log > /tmp/tf.1.log
cat /tmp/tf.1.log > ${_LOGS}/rm.logxcept.log ; fi ; fi
_XMETA="$(grep '^_XMETA' ${_LOGS}/rm.logxcept.log 2>/dev/null | sed -e 's:_XMETA:;')"
find ${_MOBAPP} -maxdepth 2 -name 'iTunesMetadadata.plist' > /tmp/tf.logrm.log
cat /tmp/tf.logrm.log |& while read _METAdt ; do find "$(echo ${_METAdt} | sed 's:^(./[-0-9]*).*:~1:~)' -name '*.app' | sed
-e 's:^.*cations:~' -e 's:~.app:~' | grep -Eww "${_XMETA}" | awk -F '/' '{printf "%s/%s\n", $2, $1}' >> /tmp/tf.tmp.log ; done
sort /tmp/tf.tmp.log 2>/dev/null | grep -Eww "${_XMETA}" | awk -F '/' '{printf "%s/%s\n", $2, $1}' > /tmp/tf.meta.log
if [ -s /tmp/tf.meta.log ] ; then printf "\n"
function imeta() { "$@" | sed 's:~.:~' | tr '~' '-' | awk '{printf "\033[36m%03d\033[0m%-19.19s\n", NR, $1}' | sed 's:^.:~' ; }
imeta cat /tmp/tf.meta.log | pa3ste
_TLOG="meta" ; editlist ; _PRINT="$(imeta cat /tmp/tf.meta.log | pa3ste)" ; editprint ; _XCPT="META" ; logxlist
grep -Fw "$_PRINT" /tmp/tf.meta.log" /tmp/tf.logrm.log | tee /tmp/tf.cout.log >> /tmp/rm.extra.log
notab echo -e "\n ${_CY}Found ${_RS}&${_CY} logged [${_WH}$(icounttmp) | sed -e "s:$(cyan)\]
$(white)iTunesMetadadata$(nocolor)\.plist:" -e '/^\<[\>\<1\>\<]\>/s/lists$/list/)'${_CY} in Mobile Apps"
else notab echo -ne "\n ${_CY}Found no '${_WH}iTunesMetadadata${_RS}.plist${_CY}' to remove ${_RS}" ; etcn ; fi ; fi
#-----
#----- Remove mobile-app cracker identification -----
#-----
find ${_MOBAPP} -mindepth 3 -maxdepth 3 -name '*.bin' -size -50c | grep '\.app' >> /tmp/rm.extra.log
#-----
#----- Remove trash leftover OS X installation files -----
#-----
cleanup ; find $Path -name '*' |& while read _TRASH ; do if grep -q 'Mac OS X' "${_TRASH}" ; then
echo $_TRASH >> /tmp/tf.logrm.log ; rm -f $_TRASH ; fi ; done
if [ -s /tmp/tf.logrm.log ] ; then bliner ; cat /tmp/tf.logrm.log >> /tmp/rm.extra.log
notab echo -e "\n ${_CY}Removed [${_WH}$(icount)${_CY}] leftover ${_RS}Mac OS${_CY} post-installation trash files" | sed
'/[^\0-9]1[^\0-9]/s/files/file/' ; fi
#-----
#----- Remove OS X ".DS_Store" files & AFP junk -----

```

```

#-----
find $Path \( -name '.AppleD*' -o -name '.DS_Store' -o -name ':2e*' \) | xargs -0 -l -d '\n' rm -rf
#-----
#----- Remove extraneous Wallpaper images -----
#-----
cleanup ; find ${_PRIVAR}/stash -type f | grep 'Wallpaper/' | grep '\.png' | grep -Ev
'default|poster|00|01|02|04|05|06|07|08|09|12|16|17|19|26' > /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then bliner
notab echo -ne "\n ${_RD}Remove unwanted stock ${_WH}Wallpaper ${_RD}images ${_RS}" ; etc ; yn ; answer ; print_yn
if [[ "$ii" = Y ]] ; then cat /tmp/tf.logrm.log >> /tmp/rm.extra.log
notab echo -ne "\n ${_CY}Removed [${_WH}$(icount | sed -e "s:$(cyan)\) stock $(nocolor)Wallpaper$(cyan) images:" -e
'/^\<[\>\<1\>\<]\>/s/s$//) from iDevice\n" ; fi ; fi
#-----
#----- Remove unused SBSetting & Winterboard Themes -----
#-----
_XTHEME="$(grep '^_XTHEME' ${_LOGS}/rm.logxcept.log 2>/dev/null | sed 's:_XTHEME::')"
declare -A _THEMES
_THEMES[SBSettingsTM]="${_MOBILE}/Library/SBSettings"
_THEMES[Winterboard]="${_PRIVAR}/stash"
for _T in "${!_THEMES[@]}" ; do if [[ $(find ${_THEMES[_T]}/*Themes* -prune 2>/dev/null) ]] ; then cleanup ; bliner
notab echo -ne "\n ${_RD}Remove unused ${_WH}${_T} ${_RD}themes ${_RS}" ; etc ; yn ; answer ; print_yn ; printf "\n"
if [[ "$ii" = N ]] ; then notab printf "${_CY}Leaving ${_RS}${_T} ${_CY}themes alone ${_RS}" ; etcn
elif [[ "$ii" = Y ]] ; then IFS=$'\n' ; find ${_THEMES[_T]}/Themes* -mindepth 1 -maxdepth 1 -type d | grep -Ev
"_XTHEMES${_XTHEME}" > /tmp/tf.logrm.log ; unset IFS
if [ -s /tmp/tf.logrm.log ] ; then
function themelist() { "$@" | sed -e 's,./,,' | awk -F '\n' '{printf "\033[36m%02d\033[0m%s\n",NR,$1}' ; }
themelist cat /tmp/tf.logrm.log
_TLOG="logrm" ; editlist ; _PRINT=$(themelist cat /tmp/tf.logrm.log) ; editprint
_XCPT="THEME" ; logxlist
notab echo -e "\n ${_CY}Marked [${_WH}$(icount | sed -e "s:$(cyan)\) $(white)${_T}$(nocolor) themes:" -e
'/^\<[\>\<1\>\<]\>/s/themes$/theme/) ${_CY} for removal"
cat /tmp/tf.logrm.log 2>/dev/null >> /tmp/rm.extra.log ; else
notab echo -ne "\r ${_CY}There were no ${_WH}${_T}${_CY} themes to remove${_RS}" ; etcn ; fi ; fi ; bottom ; fi ; done
#-----
#----- REMOVE OTHER MISCELLANEOUS EXTRANEIOUS FILES -----
#-----
find ${_MOBAPP}/*Softphone.app -type f 2>/dev/null | grep 'zip' | xargs -0 -l -d '\n' rm -f
#find ${_MOBAPP}/*SlovoEd.app -name '*.prc' -type f 2>/dev/null | grep -Ev '00A2|00A3|00A6|00A7|1A0C|202F|2068' >>
/tmp/tf.logrm.log
#-----
cleanup ; _MERC='Christmas|ColorLine|LightWood|DarkBlue|Horse|Red*SUV|Silver|Ugly|Wood|Yakuza'
if [[ ${_MOBAPP}/*Mercury3.app ]] ; then
find ${_MOBAPP}/*Mercury3.app 2>/dev/null | grep -E "${_MERC}" > /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then bliner
notab echo -ne "\n ${_CY}The app ${_WH}Mercury Browser${_CY} has hundreds of theme images which can be removed
${_RS}" ; etcn ; for E in $(echo $_MERC | tr ' ' ) ; do
notab echo -ne "\n ${_RS}Found [${_RD}$(grep -E "${E}" /tmp/tf.logrm.log | wc -l | awk '{printf "%04d",$1}' | sed -e
"s:$(nocolor)\) UI images:" -e '/^\<[\>\<1\>\<]\>/s/s$//) in Theme ${_CY}E${_RS}" | tr -d '*'
sleep 0.5 ; done
notab echo -ne "\n\n ${_WH}Remove [${_RD}$(icount | awk '{printf "%04d",$1}' | sed -e "s:$(white)\) theme images:" -e
'/^\<[\>\<1\>\<]\>/s/s$//) from [${_CY}Mercury3${_RS}.app" ; etc ; yn ; answer ; print_yn
if [[ $ii = "Y" ]] ; then cat /tmp/tf.logrm.log >> /tmp/rm.extra.log ; fi ; fi ; fi
#-----
#----- Remove over-large Retina images if smaller non-Retina image also exists -----
#-----
linar ; cleanup ; notab echo -ne "\n ${_RD}Remove very large ${_WH}Retina™ ${_RD}images ${_RS}" ; etc ; yn ; answer ;
print_yn
if [[ "$ii" = Y ]] ; then
notab printf "\033[1A ${_RS}Set ${_CY}MAX${_RS} size for ${_CY}Retina™ ${_RS}images :: recommend
${_WH}300${_RS}-${_WH}400kb \n ${_RS}As long as the smaller ${_CY}non-${_CY}Retina™ ${_RS}version ${_YL}ALSO${_RS}
exists\n The ${_WH}larger${_RS} ${_CY}Retina™${_RS} image is [${_RD}removed${_RS} to save disk space\n\n
${_RD}ENTER${_RS} ${_YL}3${_RS}-${_WH}digit [${_MG}MAXIMUM SIZE${_RS} in [${_WH}KB${_RS} [${_BL}or
${_RS}[${_CY}RETURN${_RS} to CANCEL .:" ; while read -s -n3 _SIZEKB ; do if [[ $_SIZEKB = "" ]] ; then printf "${_RD}
CANCELLED\n"
break ; elif [[ "$_SIZEKB" == [0-9][0-9][0-9] ]] ; then printf "${_RD} [$_SIZEKB]KB\n" ; break ; fi ; erata ; done
if [[ $_SIZEKB ]] ; then find ${_MOBILE}/Applications/ -name '*@2x*' -type f -size +${_SIZEKB}k 2>/dev/null | grep -E
'\.png|\.jpg|\.gif|\.tif|\.tiff' |& while read _SIZED ; do if [[ -e $(echo "$_SIZED" | sed 's:@2x:') ]] ; then echo "$_SIZED" >>
/tmp/tf.logrm.log | xargs -0 -d '\n' du -cbh 2>/dev/null | tail -1 > /tmp/tf.size.log ; fi ; done

```

```

if [ -s /tmp/tf.logrm.log ] ; then
notab echo -ne "\n ${_WH}Found [${_RD}$(icount)${_WH}] ${_CY}Retina™${_WH} images" | sed '/[^\0-9]1[^\0-9]/s/ges/ge/' ;
echo -ne " over ${_CY}${_SIZEKB}kb ${_WH}for a total of [${_RD}$(isize)${_WH}] ${_RS} \n\n ${_RS}[${_GR}0${_RS}]${_WH}Keep
${_RS}or [${_RD}1${_RS}]${_WH}Remove ${_RS}" ; etc ; oi ; zanswer ; print_oI
if [[ "$ff" = 1 ]] ; then while read -r _LINE ; do rm -f "$_LINE" ; done < /tmp/tf.logrm.log
notab echo -ne "\n ${_WH}Removed [${_RD}$(icount | sed -e "s:${_S}:(white)\]$(cyan) Retina™$(white) images:" -e
'/^\<[\>\<1\>\<]\>/s/s$/') for [${_RD}$(isize)${_WH}] from iDevice${_RS}" ; etcn ; fi ; else
notab echo -ne "\n ${_RS}Found ${_RD}no${_RS} excessively large ${_CY}Retina™${_RS} images " ; etcn ; fi ; fi
#-----
#----- Caches cleanup -----
#-----
liner ; cleanup ; notab echo -ne "\n ${_RD}Clean up ${_WH}Caches ${_RS}" ; etc ; yn ; answer ; print_yn
if [[ "$ii" = Y ]] ; then
notab echo -e "\033[1A$(blank) ${_RD}Cleaning up all ${_WH}Caches ${_RS}&${_WH} Admob${_RD} folders on iDevice"
find ${_MOBILE}/Library/Caches/ -name *Linen* | tee -a /tmp/tf.logrm.log | xargs -0 -l -d '\n' rm -f
find "${_MOBILE}/Library/Application Support" -type d | grep 'Mobclix' | tee -a /tmp/tf.logrm.log | xargs -0 -d '\n' rm -rf
2>/dev/null
_CACHES="
${_PRIVAR}/root/Library/Caches/com.saurik.Cydia/SDURLCache
${_PRIVAR}/root/Library/Caches/Snapshots
${_PRIVAR}/root/Library/Caches/com.apple.keyboards/images
${_MOBILE}/Library/Caches/admob
${_MOBILE}/Library/Caches/com.apple.AppStore
${_MOBILE}/Library/Caches/com.apple.IconsCache
${_MOBILE}/Library/Caches/com.apple.springboard.sharedimagecache/Persistent
${_MOBILE}/Library/Caches/com.apple.itunesstored
${_MOBILE}/Library/Caches/Safari/Thumbnails
${_MOBILE}/Library/Caches/Snapshots
${_MOBILE}/Library/Caches/www.gstatic.com
${_MOBILE}/Documents/admob
${_MOBAPP}/*/Library/Caches/
/Library/Caches"
for E in $_CACHES ; do
find "$E" -type f 2>/dev/null | tee -a /tmp/tf.logrm.log | xargs -0 -l -d '\n' rm -f ; done
notab echo -ne "\n ${_WH}Removed [${_RD}$(icount | sed -e "s:${_S}:(white)\] cached images:" -e
'/^\<[\>\<1\>\<]\>/s/ages$/age/') ${_RS}&${_WH} other trash from iDevice ${_RS}" ; etcn ; fi
#-----
#-----
fi
#-----
##### REMOVE FACEBOOK TWITTER + OTHER BUNDLES #####
#-----
if [[ "$(grep 'RMSOCIAL' $_LOGS)/rm.settings.log | awk -F '_' '{printf $1}'" == "YES" ]] ; then
#-----
#-----
liner ; cleanup ; notab echo -ne "\n ${_RD}Remove ${_WH}Social Networking ${_RD}bundles ${_RS}" ; etc ; yn ; answer ;
print_yn
if [[ "$ii" = Y ]] ; then
_TWITTER="$(grep '^_XREM' $_LOGS)/rm.logxcept.log 2>/dev/null | grep -o 'Twitter'"
_SOCIAL='Facebook|FBDialog|FBConnect|FacebookSDK Twitter Three20 Prime31'
_SOCREM="$(echo $_SOCIAL | sed -e 's: :|:g)"
_XREM="$(grep '^_XREM' $_LOGS)/rm.logxcept.log 2>/dev/null | sed 's:_XREM|:)"
_XKEEP="$(grep '^_XKEEP' $_LOGS)/rm.logxcept.log 2>/dev/null | sed 's:^_XKEEP|:)"
find $Path \( -name '*.bundles*' -o -name '*.framework' -o -name '*.nib' -o -name '*.sqlite' \) | grep -E "$(echo $_SOCIAL |
sed 's: :|:g)" | egrep -iv 'Iconography|dpkg|Cache' > /tmp/tf.social.log
if [ -s /tmp/tf.social.log ] ; then
notab echo -e "\033[1A ${_RD}Select ${_WH}Social Networking ${_RD}bundles to mark for removal ${_RS}: "
for E in $_SOCIAL ; do
if [[ $(echo $_XKEEP | grep -Eo "${E%|*}") ]] ; then
notab printf "\n ${_CY}Found '${_WH}${E%|*}'${_CY}' on the ${_WH}Protected${_CY} list" ; sleep 3
notab printf "\r ${_MG}Remove protected status ${_WH}&${_MG} find ${_WH}${E%|*}' ${_MG}bundles ${_RS}" ; etc ; yn
answer ; print_yn ; sleep 0.5 ; if [[ "$ii" = Y ]] ; then blank ; _XKEEP="$(printf "$_XKEEP" | tr '|' '\n' | grep -v "${E}" | sort -fu |
tr '\n' '|' | sed 's:|:|:)"
updatxcept ; ii="Y" ; fi ; elif [[ $(echo $_XREM | grep -Eo "${E%|*}") ]] ; then
notab printf "\n ${_MG}Already marked '${_RS}${E%|*}'${_MG}' for removal ${_RS}" ; etcn ; sleep 1 ; blank ; bottom ; ii="Y"
else notab echo -ne "\n ${_RS}Remove '${_RD}${E%|*}'${_RS}'\t$(repeat 6 printf '\b') bundles " ; etc ; yn ; answer ; print_yn ;
fi
if [[ "$ii" = N ]] ; then notab echo -e "\033[1A$(blank) ${_YL}Skipping ${_WH}${E%|*}'\t$(repeat 6 printf '\b')${_YL} bundles"

```

```

_XKEEP="$xkeep" ; updatxcpt ; elif [ [ "$ii" = Y ] ] ; then
_XREM="$xrem" ; updatxcpt ; cat /tmp/tf.social.log 2>/dev/null | grep -Ew "${E}" > /tmp/tf.logrm.log
if [ -s /tmp/tf.logrm.log ] ; then
notab echo -e "\033[1A$(blank) ${_CY}Found ${_RS}&${_CY} logged [${_WH}$(icount | sed -e "s:$(cyan) $(white)${E%*}\|*}
$(cyan)bundles:" -e '/^\<[\>\<1\>\<]\>/s/bundles/bundle/)"
cat /tmp/tf.logrm.log >> /tmp/rm.social.log ; else
notab printf "\033[1A$(blank) ${_BL}Found no ${_RS}${E%*}\|*}\t$(repeat 6 printf '\b')${_BL} bundles\n" ; fi ; fi ; done ; else
notab echo -e "\n ${_BL}Found no ${_RS}Social Networking ${_BL}bundles" ; fi ; fi
#-----
#-----
fi
#-----
##### REPAIR PERMISSIONS #####
#-----
if [ [ "$(grep 'PERMISSIONS' ${_LOGS}/rm.settings.log | awk -F '_' '{printf $1}' )" == "YES" ] ] ; then
#-----
#-----
liner ; notab echo -e "\n ${_RD}Verifying correct $(repeat 8 printf "${_BK}")${_WH}ownerships${_RD} &
${_WH}permissions${_RD} on iDevice"
#-----
#----- Fix ownerships for Mobile, Wheel, and Admin -----
#-----
declare -a _GROUP
_GROUP[0]="${_MOBAPP} ${_MOBILE}/Media ${_MOBILE}/Library ${_DOWNL}/Iconography/Mobile_"
_GROUP[1]="${_SYSLIB}/CoreServices/SpringBoard.app ${_SYSLIB}/Frameworks/UIKit.framework ${_SYSLIB}/LaunchDaemons
${_SYSLIB}/PreferenceBundles ${_SYSLIB}/WeeAppPlugins /Library/MobileSubstrate /Library/PreferenceBundles
/Library/PreferenceLoader /Library/WeeLoader ${_PRIVAR}/stash/include ${_PRIVAR}/stash/Themes* /Library/Activator
${_MOBILE}/Library/SBSettings/Toggles ${_MOBILE}/Library/SBSettings/Commands ${_DOWNL}/Iconography/Library_
${_DOWNL}/Iconography/System_ ${_DOWNL}/Iconography/Audio_ ${_DOWNL}/Iconography/Extra_
${_PRIVAR}/stash/share/ultrasn0w /usr"
_GROUP[2]="/Library/Application*Support /Library/Frameworks ${_PRIVAR}/stash/Applications*
${_DOWNL}/Iconography/Stash_"
_ID=('mobile:mobile' 'root:wheel' 'root:admin')
_XCHWN=('SBSettings|iFile|Caches|Stash_|System_|Library_|Extra_|Audio_|' 'Caches' 'Caches')
liner ; cleanup
for E in ${!_ID[@]} ; do
notab echo -ne "\n ${_CY}Verifying [${_WH}$(echo ${_ID[E]} | awk -F : '{printf "%s\\\"${_CY}:'\\\"${_WH}'%s", $1, $2)'}${_CY}
ownerships ${_RS}" ; etcn ; gap
for D in ${_GROUP[E]} ; do
if [ [ -e $D ] ] ; then
find "$D" \(! -user "${_ID[E]:*}" -o ! -group "${_ID[E]:*}" \) 2>/dev/null | grep -E -v "${_XCHWN[E]}" | tee -a
"/tmp/tf.${_ID[E]:*.log" | xargs -0 -l -d '\n' --no-run-if-empty chown -h ${_ID[E]}
if [ [ $(grep "^$D" "/tmp/tf.${_ID[E]:*.log") ] ] ; then
notab echo -ne "\n ${_RS}[${_RD}+${_RS}] ${_RD}Repairing [${_WH}$(grep "^$D" /tmp/tf.${_ID[E]:*.log | wc -l | awk '{print
"%04d", $1}' | sed -e "s:$(red)\| files:" -e '/^\<[\>\<1\>\<]\>/s/files/ file /) in ${_RS}$(shortd echo -e "$D")\n"
redgap ; else notab echo -ne "\n ${_RS}[${_CY}~/${_RS}] ${_RS}$(shortd echo -e "$D")\n" ; redgap ; fi ; fi ; done ; done
#-----
#----- Fix Symlink directory ownerships -----
#-----
find ${_MOBILE} -maxdepth 2 -type l \(! -user "mobile" -o ! -group "mobile" \) 2>/dev/null | tee -a /tmp/tf.chown.log |
xargs -0 -l -d '\n' --no-run-if-empty chown -h mobile:mobile
find ${_STAPPS} -maxdepth 2 -type l \(! -user "root" -o ! -group "admin" \) 2>/dev/null | tee -a /tmp/tf.chown.log | xargs
-0 -l -d '\n' --no-run-if-empty chown -h root:admin
find /Library -maxdepth 1 -type l \(! -user "root" -o ! -group "admin" \) 2>/dev/null | tee -a /tmp/tf.chown.log | xargs -0
-l -d '\n' --no-run-if-empty chown -h root:admin
#-----
#----- Fix executable permission -----
#-----
_DIR="/System/Library/PreferenceBundles /Library/PreferenceBundles"
bliner ; notab echo -ne "\n ${_CY}Verifying correct [${_WH}755${_CY}] permissions for PreferenceBundle ${_WH}Executables
${_RS}" ; etcn ; for E in $_DIR ; do cleanup
find $E -mindepth 2 -type f ! -name ".*" ! -perm 0755 | grep -Ev 'CodeResources' | tee -a /tmp/tf.chmod.log | xargs -0 -l
-d '\n' --no-run-if-empty chmod 755
if [ -s /tmp/tf.chmod.log ] ; then
notab echo -e "\n ${_RS}[${_RD}+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}$(grep -E "^$E" /tmp/tf.chmod.log |
wc -l | sed -e "s:$(nocolor)\| executables:" -e '/^\<[\>\<1\>\<]\>/s/les/le/)' in $(white)$(echo "$E" | sed -e "s:/:/:" -e
"s:/:$(red)\|$(white):g")\n"
awk '{printf "\033[36m%02d\033[0m%-s\n", NR, $0}' /tmp/tf.chmod.log | sed -e "s:/:/:" -e

```

```

"s: \(settings\|preferences*\)\):$(blue)~$(nocolor):!" -e 's:^\:t:'
else notab echo -e "\n ${_RS}${_CY}√${_RS}] ${_GR}No repairs needed${_RS} in ${_WH}$(echo ${E} | sed -e 's:^\:/' -e
"s:/:$(red)\:$(white):g") ${_RS}"
fi ; done
#-----
#----- Fix directory permissions in 'mobile' -----
#-----
bliner ; notab echo -ne "\n ${_CY}Verifying correct [${_WH}755${_CY}] permissions for ${_WH}Directories ${_RS}" ; etcn ;
cleanup
find ${_MOBILE} -type d ! \( -perm 0755 -o -perm 0775 \) | grep -v 'Cache' | tee -a /tmp/tf.chmod.log | xargs -0 -l -d '\n'
--no-run-if-empty chmod 755
if [ -s /tmp/tf.chmod.log ] ; then cat /tmp/tf.chmod.log > /tmp/tf.logrm.log
notab echo -e "\n ${_RS}${_RD}+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}$(icount | sed -e "s:/:$(nocolor)\]
directories:" -e '/^\<[\>\<1\>\<]\>/s/ries/ry/') in ${_WH}mobile${_RS}"
else notab echo -e "\n ${_RS}${_CY}√${_RS}] ${_WH}Directory ${_RS}permissions needed no modifications" ; fi
#-----
#----- Fix MobileSubstrate Dylib permissions -----
#-----
notab echo -ne "\n ${_CY}Verifying correct [${_WH}755${_CY}] permissions for ${_WH}Dynamic Libraries ${_RS}" ; etcn ;
cleanup
find $Path \( -name '*.dylib' -o -name '*.sh' \) -type f ! \( -perm 0755 -o -perm 0775 \) | tee -a /tmp/tf.chmod.log | xargs
-0 -l -d '\n' --no-run-if-empty chmod 755
if [ -s /tmp/tf.chmod.log ] ; then cat /tmp/tf.chmod.log > /tmp/tf.logrm.log
notab echo -e "\n ${_RS}${_RD}+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}$(icount | sed -e "s:/:$(nocolor)\]
$(white)dylib$(nocolor) extensions:" -e '/^\<[\>\<1\>\<]\>/s/ions/ion/') in ${_WH}Library${_RS}"
else notab echo -e "\n ${_RS}${_CY}√${_RS}] ${_WH}dylib${_RS} permissions needed no modifications" ; fi
#-----
#----- Find incorrect ((( 600 / 644 / 664 ))) permissions system-wide -----
#-----
find $Path \( -name "CodeResources" -o -iname "PkgInfo" -o -iname "*\.mobileprovision" -o -iname "*\.example" -o
-iname "*\.plist" -o -iname "*\.string" -o -iname "*\.nib" -iname "*.caf" -o -iname "*\.wav" -o -iname "*\.aif*" -o -iname
"*\.m4[ra]" -o -iname "*\.mp[34]" -iname "*\.html" -o -iname "*\.txt" -o -iname "*\.log" -o -iname "*\.db" -o -iname
"*\.sqlitedb" -iname "*\.png" -o -iname "*\.jpe*g" -o -iname "*\.tif*" \) -type f ! \( -perm 0600 -o -perm 0644 -o -perm
0664 \) > /tmp/chmod.log
#-----
#----- PLIST - STRINGS - NIB - Fix permissions system-wide -----
#-----
notab echo -ne "\n ${_CY}Verifying correct [${_WH}644${_CY}] permissions for ${_WH}Property Lists ${_RS}" ; etcn ; cleanup
grep -i 'plist$\|strings$\|nib$' /tmp/chmod.log | tee /tmp/tf.tmp.log | xargs -0 -l -d '\n' --no-run-if-empty chmod 644
if [ -s /tmp/tf.tmp.log ] ; then cat /tmp/tf.tmp.log > /tmp/tf.logrm.log
notab echo -e "\n ${_RS}${_RD}+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}$(icount | sed -e "s:/:$(nocolor)\]
$(white)plist$(nocolor)/$(white)strings$(nocolor)/$(white)nib$(nocolor) files:" -e '/^\<[\>\<1\>\<]\>/s/files/file/')
system-wide" ; else notab echo -e "\n ${_RS}${_CY}√${_RS}] ${_WH}plist${_RS}, ${_WH}strings ${_RS}&${_WH} nib${_RS}
permissions needed no modifications" ; fi
#-----
#----- PkgInfo - CodeResources - Fix permissions system-wide -----
#-----
notab echo -ne "\n ${_CY}Verifying correct [${_WH}644${_CY}] permissions for ${_WH}Resource files ${_RS}" ; etcn ; cleanup
grep -i 'CodeResources\|PkgInfo\|\.mobileprovision\|\.example' /tmp/chmod.log | tee /tmp/tf.tmp.log | xargs -0 -l -d '\n'
--no-run-if-empty chmod 644
if [ -s /tmp/tf.tmp.log ] ; then cat /tmp/tf.tmp.log > /tmp/tf.logrm.log
notab echo -e "\n ${_RS}${_RD}+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}$(icount | sed -e "s:/:$(nocolor)\]
$(white)Resource $(nocolor)files:" -e '/^\<[\>\<1\>\<]\>/s/files/file/') system-wide" ; else notab echo -e "\n
${_RS}${_CY}√${_RS}] ${_WH}Resource file${_RS} permissions needed no modifications" ; fi
#-----
#----- CAF - WAV - MP3 - MP4 - M4A - M4P - AIFF - Fix permissions system-wide -----
#-----
notab echo -ne "\n ${_CY}Verifying correct [${_WH}644${_CY}] permissions for ${_WH}Audio Files ${_RS}" ; etcn ; cleanup
grep -i 'caf$\|wav$\|aif*$\|m4[ra]$\|mp[34]' /tmp/chmod.log | tee /tmp/tf.tmp.log | xargs -0 -l -d '\n'
--no-run-if-empty chmod 644
if [ -s /tmp/tf.tmp.log ] ; then cat /tmp/tf.tmp.log > /tmp/tf.logrm.log
notab echo -e "\n ${_RS}${_RD}+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}$(icount | sed -e "s:/:$(nocolor)\]
$(white)audio$(nocolor) files:" -e '/^\<[\>\<1\>\<]\>/s/files/file/') system-wide" ; else notab echo -e "\n
${_RS}${_CY}√${_RS}] ${_WH}audio file${_RS} permissions needed no modifications" ; fi
#-----
#----- HTML - TXT - LOG - SQLITEDB - DB - Fix permissions system-wide -----
#-----
notab echo -ne "\n ${_CY}Verifying correct [${_WH}644${_CY}] permissions for ${_WH}TXT ${_RS}&${_WH} HTML

```



```

${_RS}&${_WH} LOG ${_RS}&${_WH} DATABASE Files ${_RS}"; etcn ; cleanup
grep -i 'html$\|txt$\|log$\|db$\|sqlitedb' /tmp/chmod.log | tee /tmp/tf.tmp.log | xargs -0 -l -d '\n' --no-run-if-empty
chmod 644
if [ -s /tmp/tf.tmp.log ] ; then cat /tmp/tf.tmp.log > /tmp/tf.logrm.log
notab echo -e "\n ${_RS}[${_RD}]+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}]$(icount | sed -e "s:$(nocolor)\|
$(white)html$(nocolor)/$(white)txt$(nocolor)/$(white)log$(nocolor)/$(white)database$(nocolor) files:" -e
'/^\|<[\>\<1\>\<]\>/s/files/file/')"; else
notab echo -e "\n ${_RS}[${_CY}~${_RS}] ${_WH}txt${_RS}, ${_WH}html${_RS}, ${_WH}log ${_RS}&${_WH} database file
${_RS}permissions needed no modifications" ; fi
#-----
#----- PNG - JPG Fix permissions system-wide ** First time will take forever ** -----
#-----
notab echo -ne "\n ${_CY}Verifying correct [${_WH}644${_CY}] permissions for [${_WH}]PNG ${_RS}&${_WH} JPG Images ${_RS}"
; etcn ; cleanup
grep -i 'png$\|jpg[e]*g$\|tif*$' /tmp/chmod.log | tee /tmp/tf.tmp.log | xargs -0 -l -d '\n' --no-run-if-empty chmod 644
if [ -s /tmp/tf.tmp.log ] ; then cat /tmp/tf.tmp.log > /tmp/tf.logrm.log
notab echo -e "\n ${_RS}[${_RD}]+${_RS}] ${_RD}Repaired permissions${_RS} on [${_WH}]$(icount | sed -e "s:$(nocolor)\|
$(white)png$(nocolor)/$(white)jpg$(nocolor) images:" -e '/^\|<[\>\<1\>\<]\>/s/mages/mage/')"
else notab echo -e "\n ${_RS}[${_CY}~${_RS}] ${_WH}png ${_RS}&${_WH} jpg image ${_RS}permissions needed no
modifications" ; fi
#-----
#----- Fix SBSettings permissions in 'mobile/Library' -----
#-----
chmod 777 ${_MOBILE}/Library/SBSettings
chmod -R 755 ${_MOBILE}/Library/SBSettings/Commands
#-----
#-----
fi
#-----
##### REMOVE NON-RETINA IMAGES - FOR 4G & 4Gs DEVICES ONLY #####
#-----
if [[ "$(grep 'RMPNG' ${_LOGS}/rm.settings.log | awk -F ' ' '{printf $1}')" == "YES" ]] ; then
#-----
#-----
liner ; cleanup ; notab echo -ne "\n ${_RD}Remove [${_WH}]non${_RS}-[${_WH}]Retina™[${_RD}] images [${_RS}"; etc ; yn ; answer
; print_yn
if [[ "$ii" = Y ]] ; then
notab printf "\033[1A$(blank) ${_RS}If your iDevice is 4G or newer then [${_CY}]it has a Retina™ hi-res display\n [${_RS}]Often
apps come bundled w/both [${_RD}]Retina™ [${_RS}&${_RD}] non-Retina™ [${_RS}]image versions\n [${_RS}]To make them
compatible with either older & newer iDevices; however\n This means that many duplicate images are eating up your disk
space\n 4G+ iDevices can safely remove [${_RD}]non-Retina™[${_RS}] image versions, provided\n The [${_MG}]Retina™
[${_RS}]version also exists " ; etcn
notab echo -ne "\n [${_WH}]Searching for [${_RD}]non-Retina™ [${_WH}]images [${_NL}]with [${_MG}]Retina™ [${_WH}]counterparts
[${_RS}"; etcn ; bliner
find $Path \( -iname '*.png' -o -iname '*.jpg' \) -type f | grep -Ev
'Iconography|Caches|Trash|TimeCapsule|tmp|iFile.app|Icon-Small' | sed 's:@2x::' | uniq -d | tee -a /tmp/rm.regular.log |
xargs -0 -d '\n' du -cbh 2>/dev/null | tail -1 > /tmp/tf.size.log
cat /tmp/rm.regular.log | sed 's:(.*)/:1%:' | tee -a /tmp/tf.logrm.log | sed 's,[^%]*$,,' | uniq |&
while read _iDIR ; do
notab echo -ne "\n [${_RD}]Found [${_WH}]$(grep -c "${_iDIR}" /tmp/tf.logrm.log | awk '{printf "%04d", $0}' | sed -e
"s:$(red)\| non-Retina™ images:" -e '/0001/s/mages/mage/') in [${_NL}]$(dirname "${_iDIR}" | sed -e 's:/:/' -e 's:%$::' -e
's: :g' -e 's:(.|\{3\})$:%%\1:' | awk 'BEGIN { RS = "/" } ; { printf "\033[37m%.4s\033[31m:.", $0 }' | sed
's:%%$::)[${_WH}]$(bname basename "${_iDIR}")${_RD}]\n" | tee -a /tmp/tf.regular.log ; echo >> /tmp/tf.regular.log ; shortpath
grep "${_iDIR}" /tmp/tf.logrm.log >> /tmp/tf.regular.log ; done
if [[ -s /tmp/tf.logrm.log ]] ; then echo "$(bliner)$(sfix)$(bliner)" >> /tmp/tf.regular.log
notab echo -e "\n [${_WH}]System${_RS}-[${_WH}]wide total [${_MG}]non-Retina™[${_WH}] images that have [${_RD}]Retina™[${_WH}]
counterparts [${_RD}]$(icount)${_WH}] for [${_RD}]$(isize)${_WH}]"
notab echo -ne "\n [${_RS}]List all [${_WH}]non${_RS}-[${_WH}]Retina™[${_RS}] images with [${_WH}]Retina™
[${_RS}]counterparts[${_RS}"; etc ; yn ; answer ; print_yn ; if [[ "$ii" = Y ]] ; then cat /tmp/tf.regular.log ; fi
else notab echo -ne "\n [${_RS}]Found no trashable [${_MG}]non-Retina™[${_RS}] images " ; etcn ; fi ; fi
#-----
#-----
fi
#-----
##### REMOVE RETINA IMAGES - FOR 2G, 3G & 3Gs DEVICES ONLY #####
#-----
#-----
if [[ "$(grep '2xPNG' ${_LOGS}/rm.settings.log | awk -F ' ' '{printf $1}')" == "YES" ]] ; then

```

```
#-----
#-----
liner ; cleanup ; notab echo -ne "\n ${_RD}Remove all ${_WH}Retina™${_RD} images ${_RS}" ; etc ; yn ; answer ; print_yn
if [[ "$ii" = Y ]] ; then
notab printf "\033[1A$(blank) ${_RS}If your iDevice isn't 4G or newer ${_CY}it doesn't have a Retina™ display\n ${_RS}Often
apps come bundled w/both ${_RD}Retina™ ${_RS}&${_RD} non-Retina™ ${_RS}image versions\n ${_RS}To make them
compatible with either older & newer iDevices; however\n This means that many duplicate images are eating up your disk
space\n 2G & 3G(s) iDevices can safely remove all ${_RD}Retina™${_RS} hi-res images " ; etcn
notab echo -ne "\n ${_WH}Searching iDevice for ${_CY}Retina™${_WH} images ${_RS}" ; etcn ; bliner
find $Path -iname *@2x* -type f | grep -Ev 'Iconography|Caches|Trash|TimeCapsule|tmp' | tee -a /tmp/rm.retina.log |
xargs -0 -d '\n' du -cbh 2>/dev/null | tail -1 > /tmp/tf.size.log
cat /tmp/rm.retina.log | sed 's:\(.*\)\/:1%:' | tee -a /tmp/tf.logrm.log | sed 's,[^%]*$,,' | uniq |&
while read _iDIR ; do
notab echo -ne "\n ${_RD}Found [${_WH}$(grep -c "$_iDIR" /tmp/tf.logrm.log | awk '{printf "%04d", $0}' | sed -e
"s:~:~(red)\} Retina™ images:" -e '/0001/s/mages/mage /') in $_NL]$(dirname "$_iDIR" | sed -e 's:/:/' -e 's:%$::' -e 's:~:~g'
-e 's:/\(\.\{3\}\)$:/%%\1:' | awk 'BEGIN { RS = "/" } ; { printf "\033[37m%.4s\033[31m:~:~$0 }' | sed 's:~:~:~)${_WH}$(bname
basename "$_iDIR")${_RD}]\n" | tee -a /tmp/tf.retina.log ; echo >> /tmp/tf.retina.log ; shortpath grep "$_iDIR"
/tmp/tf.logrm.log >> /tmp/tf.retina.log ; done
if [[ -s /tmp/tf.logrm.log ]] ; then echo "$(bliner)$(sfix)$(bliner)" >> /tmp/tf.retina.log
notab echo -e "\n ${_WH}Total ${_CY}Retina™${_WH} images system${_RS}-${_WH}wide [${_RD}$(icount)${_WH}] for
[${_RD}$(isize)${_WH}]"
notab echo -ne "\n ${_RS}List all ${_WH}Retina™${_RS} images${_RS}" ; etc ; yn ; answer ; print_yn ; if [[ "$ii" = Y ]] ; then cat
/tmp/tf.retina.log ; fi
else notab echo -ne "\n ${_RS}Found no trashable ${_MG}Retina™${_RS} images " ; etcn ; fi ; fi
#-----
#-----
fi
#-----
#-----
##### REMOVE iPad-SPECIFIC FILES #####
#-----
#-----
if [[ "$(grep 'RMIPAD' ${_LOGS}/rm.settings.log | awk -F '_' '{printf $1}')" == "YES" ]] ; then
#-----
#-----
liner ; cleanup ; notab echo -ne "\n ${_RD}Remove ${_WH}iPad${_RS}-${_WH}specific${_RD} files ${_RS}" ; etc ; yn ; answer ;
print_yn
if [[ "$ii" = Y ]] ; then
notab printf "\033[1A$(blank) ${_WH}Searching for ${_RD}iPad${_RS}-${_WH}specific files to remove ${_RS}" ; etcn ; bliner ;
sleep 1
find $Path -iname *iPad* -type f | grep -E '.png|.jpg|.plist|.string|.nib|.html|.xml' | grep -Ev
'Iconography|Caches|Trash|TimeCapsule|tmp' | tee -a /tmp/rm.ipad.log | xargs -0 -d '\n' du -cbh 2>/dev/null | tail -1 >
/tmp/tf.size.log
cat /tmp/rm.ipad.log | sed 's:\(.*\)\/:1%:' | tee -a /tmp/tf.logrm.log | sed 's,[^%]*$,,' | uniq |& while read _iDIR ; do
notab echo -ne "\n ${_RD}Found [${_WH}$(grep -c "$_iDIR" /tmp/tf.logrm.log | awk '{printf "%04d", $0}' | sed -e
"s:~:~(red)\} iPad files:" -e '/0001/s/files/file /') in $_NL]$(dirname "$_iDIR" | sed -e 's:/:/' -e 's:%$::' -e 's:~:~g' -e
's:/\(\.\{3\}\)$:/%%\1:' | awk 'BEGIN { RS = "/" } ; { printf "\033[37m%.4s\033[31m:~:~$0 }' | sed 's:~:~:~)${_WH}$(bname
basename "$_iDIR")${_RD}]\n" | tee -a /tmp/tf.ipad.log ; echo >> /tmp/tf.ipad.log ; shortpath grep "$_iDIR"
/tmp/tf.logrm.log >> /tmp/tf.ipad.log ; done
if [[ -s /tmp/tf.logrm.log ]] ; then echo "$(bliner)$(sfix)$(bliner)" >> /tmp/tf.ipad.log
notab echo -e "\n ${_WH}Total removable ${_MG}iPad${_WH} files system${_RS}-${_WH}wide [${_RD}$(icount)${_WH}] for
[${_RD}$(isize)${_WH}]"
notab echo -ne "\n ${_RS}List all ${_WH}iPad${_RS}-${_WH}specific${_RS} files${_RS}" ; etc ; yn ; answer ; print_yn ; if [[ "$ii" =
Y ]] ; then cat /tmp/tf.ipad.log ; fi
else notab echo -ne "\n ${_RS}Found no trashable ${_MG}iPad${_RS} files " ; etcn ; fi ; fi
#-----
#-----
fi
#-----
#-----
##### REMOVE iPhone 5 TALL RETINA IMG FILES #####
#-----
#-----
if [[ ! "$(grep 'i5G' ${_LOGS}/rm.settings.log)" ]] ; then
#-----
#-----
liner ; cleanup ; notab echo -ne "\n ${_RD}Remove ${_WH}iPhone5 Tall${_RS}-${_WH}Retina™${_RD} images ${_RS}" ; etc ; yn ;
answer ; print_yn
if [[ "$ii" = Y ]] ; then
```

```

notab printf "\033[1A$(blank) ${_WH}Searching for ${_MG}iPhone5${_RD} Tall${_RS}-${_RD}Retina™${_WH} images to remove
${_RS}"; etcn
bliner
find $Path -iname *-568h* -type f | tee /tmp/rm.ip5.log | grep -Ev 'Iconography|Caches|Trash|TimeCapsule|tmp' | xargs -0
-d '\n' du -cbh 2>/dev/null | tail -1 > /tmp/tf.size.log
cat /tmp/rm.ip5.log | sed 's:^\(.*\):\1%:' | tee -a /tmp/tf.logrm.log | sed 's,[^%]*$,,' | uniq |& while read _iDIR; do
notab echo -ne "\n ${_RD}Found [${_WH}$(grep -c "${_iDIR}" /tmp/tf.logrm.log | awk '{printf "%03d", $0}')] images:" -e '/001/s/ages/age /' in ${_NL}[${_iDIR} | sed -e 's:^\(:.*$)' -e
's:%$:' -e 's: ::g' -e 's: /\(\.\{3\}\)$:/%%%1:' | awk 'BEGIN { RS = "/" }; { printf "\033[37m%.4s\033[31m: ", $0 } | sed
's:%$:::${_WH}$(bname basename "${_iDIR}")${_RD}]\n' | tee -a /tmp/tf.ip5.log ; echo >> /tmp/tf.ip5.log ; shortpath grep
"${_iDIR}" /tmp/tf.logrm.log >> /tmp/tf.ip5.log ; done
if [[ -s /tmp/tf.logrm.log ]] ; then echo "${bliner}$(sfix)$(bliner)" >> /tmp/tf.ip5.log
notab echo -e "\n ${_WH}Total removable ${_MG}iPhone5${_WH} images system${_RS}-${_WH}wide [${_RD}$(icount)${_WH}
for [${_RD}$(isize)${_WH}]
notab echo -ne "\n ${_RS}List all ${_WH}iPhone5${_RS}-${_WH}specific${_RS} images${_RS} " ; etc ; yn ; answer ; print_yn ; if
[[ "$ii" = Y ]] ; then cat /tmp/tf.ip5.log ; fi
else notab echo -ne "\n ${_RS}Found no trashable ${_MG}iPhone${_RS} files " ; etcn ; fi ; fi
#-----
#-----
fi
#-----
##### DELETE FILES IN REMOVE-LOGS #####
#-----
#-----
repeat 9 dot
#-----
function rmalias() { echo "$E" | sed -e 's:proj:Language Packs:' -e 's:carrier:Carrier Bundles:' -e 's:txinput:TextInput
Bundles:' -e 's:keyboard:Keyboard Layouts:' -e 's:extra:Miscellaneous & Junk:' -e 's:social:Social Bundles:' -e
's:regular:non~Retina™ images:' -e 's:retina:Retina™ images:' -e 's:ipad:iPad images:' -e 's:ip5:Tall~Retina™ images:' ; }
#-----
function rrm() { notab echo -ne "\n ${_YL}[!${_RD}] REMOVING ${_RS}" ; etc ; while read -r _LINE ; do rm -fr "$_LINE" ; done
2>/dev/null < /tmp/rm.${E}.log ; cat /tmp/rm.${E}.log 2>/dev/null >> /tmp/rm.removed.log ; printf "$(repeat ${$(tput
cols)/2} printf '_'\n\n" >> /tmp/rm.removed.log ; etc ; echo -e " ${_WH}DONE ${_YL}[!${_RS}" ; rm /tmp/rm.${E}.log
2>/dev/null ; }
#-----
_RMLOG='proj carrier txinput keyboard extra social regular retina ipad ip5'
for _IF in $_RMLOG ; do
if [ -s /tmp/rm.${_IF}.log ] ; then _TEST="YES" ; fi ; done
if [[ "$_TEST" = YES ]] ; then liner
echo -e "\033[1A\r\t\t\t\t\t\b\b\b${_WH}REVIEW${_RD}-${_RS}&${_RD}-${_WH}REMOVE${_RD}-${_WH}FILES${_RS}"
notab echo -ne "\n ${_RS}You can review each removal list before marked files are actually deleted\n ${_RD}Proceed with
care${_RS}: if in doubt, select ${_WH}[0]${_RS} to ${_YL}CANCEL${_RS} & check the logfiles\n SSH into iDevice or use
${_MG}iFile${_RS} to find logs in ${_WH}mobile${_RD}:${_WH}Documents${_RD}:${_WH}logs${_RD}:${_WH}saved ${_RS}\n"
bliner ; for E in $_RMLOG ; do if [ -s /tmp/rm.${E}.log ] ; then
notab echo -ne "\n ${_RS}===${_YL}$(repeat 36 printf '='${_RS}===\r\t${_YL}== ${_WH}$(rmalias | tr '[a-z]' '[A-Z]') | sed -e
's:\&:$(nocolor)\&$(white):" -e 's:\-:$(nocolor)\-$(white):" )\n"
notab echo -ne "\n ${_GR}[CANCEL]${_WH}[0]${_RD}]AND SAVE THIS LOGFILE\n ${_YL}[DELETE]${_WH}[1]${_RD}]ALL FILES IN
THE LIST\n ${_CY}[REVIEW]${_WH}[9]${_RD}]ITEMS IN REMOVAL LIST ${_RS}"
etc ; ask ; while read -s -n 1 tt ; do if [[ "$tt" = 9 ]] ; then printf " ${_RS}[${_CY}REVIEW${_RS}]\n"
notab echo -e "\n ${_YL}[!${_CY}] Reviewing ${_WH}$(rmalias | sed -e 's:\&:$(nocolor)\&$(white):" -e
's:\-:$(nocolor)\-$(white):")${_CY} removal list "${_YL}[!${_RS}]\n"
function rmlist() { shortd "$@" | awk '{printf "\033[36m%03d\033[31m:\033[0m%-s\n",NR,$0}' ; }
rmlist cat /tmp/rm.${E}.log ; cat /tmp/rm.${E}.log > /tmp/tf.logrm.log
_TLOG="logrm" ; editlist ; _PRINT="$(rmlist cat /tmp/tf.logrm.log)" ; editprint
cat /tmp/tf.logrm.log > /tmp/rm.${E}.log
notab echo -ne "\n ${_YL}[!${_WH}] Remove listed ${_CY}$(rmalias | sed -e 's:\&:$(nocolor)\&$(cyan):" -e
's:\-:$(nocolor)\-$(cyan):" ) ${_WH}files ${_RS}" ; etc ; yn ; answer ; print_yn ; if [[ "$ii" = Y ]] ; then _RMV="YES" ; fi ; break
elif [[ "$tt" = 1 ]] ; then _RMV="YES" ; printf " ${_RS}[${_YL}DELETE${_RS}]\n" ; break
elif [[ "$tt" = 0 ]] ; then printf " ${_RS}[${_GR}CANCEL${_RS}]\n" ; break ; fi ; erata ; done
if [[ "$_RMV" == "YES" ]] ; then rrm ; notab echo -e "\n ${_RS}===${_YL}$(repeat 36 printf '='${_RS}===
else cat /tmp/rm.${E}.log 2>/dev/null > ${_LOGS}/saved/rm.${E}.log
notab echo -ne "\n ${_RS}===${_RD}$(repeat 36 printf '='${_RS}===\r\t\t\t ${_WH}STOPPED \n\n ${_CY}Saving
${_WH}$(rmalias | sed -e 's:\&:$(nocolor)\&$(white):" -e 's:\-:$(nocolor)\-$(white):" ) ${_CY}Remove-List to\n\n
${_RS}private${_RD}:${_RS}var${_RD}:${_RS}mobile${_RD}:${_RS}Documents${_RD}:${_RS}saved${_RD}:${_RS}logs\n\n
${_YL}You can run through it again later ${_RS}" ; etcn ; echo -e "\n ${_RS}===${_RD}$(repeat 36 printf '='${_RS}===
${_RS}" ; fi ; fi ; done ; else liner ; bottom
notab echo -ne "\n ${_WH}There are ${_RD}no files to delete" ; cutxt " ${_WH}so you must be ${_CY}done${_RS} " ; etcn ; liner
; fi

```

```
#-----
clear ; repeat 15 dot ; notab echo -e "\n
${_RS}---${_RD}---${_RS}THANK${_RD}-${_RS}YOU${_RD}-${_RS}DAVE${_RD}--${_RS}[${_RD}√${_RS}]${_RD}--${_RS}HAVE${_
RD}-${_RS}A${_RD}-${_RS}NICE${_RD}-${_RS}DAY${_RD}---${_RS}---\n" ; sleep 1
#-----
if [[ "$_REMLOG" == "YES" ]] ; then rm -f ${_LOGS}/rm.settings.log 2>/dev/null ; fi
#-----
#####
#-----
repeat 20 dot ; sleep 3 ; clear ; notab echo -e "\n
${_RS}---${_RD}---${_RS}END${_RD}-${_RS}OF${_RD}-${_RS}PROCESS${_RD}-&-${_RS}EXIT${_RD}-${_RS}STAGE${_RD}-${_RS}
LEFT${_RD}----${_RS}---\n\n"
#-----
#-----
exitxt
```