

```
#!/bin/bash
# build_repo.sh - Build a custom pacman repository.
# Features: Steampunk theme with automatic Lolcat Rainbow fallback for
headers.
```

```
# --- Steampunk Color Definitions (256-Color) ---
export NC='\033[0m'
export STEAM_WHITE='\033[38;5;253m'      # Steam / Fog
export STEAM_BRONZE='\033[38;5;136m'     # Dark Gold/Bronze
export STEAM_COPPER='\033[38;5;172m'     # Copper/Orange
export STEAM_GOLD='\033[38;5;220m'       # Bright Brass/Gold
export STEAM_IRON='\033[38;5;245m'       # Iron/Gray
export STEAM_RUST='\033[38;5;130m'       # Rust/Leather
export STEAM_GEAR='\033[38;5;240m'       # Dark Gear Gray
```

```
# --- Configuration ---
REPO_DIR="./my_custom_repo"
REPO_NAME="my_custom_repo"
AUR_LIST="data/aur.list"
PKG_LIST="data/packages.list"
BUILD_DIR="/tmp/repo_builder"
```

```
# --- Visual Functions ---
```

```
# Helper to print text, optionally piping through lolcat if available
print_rainbow() {
    if command -v lolcat &> /dev/null; then
        echo -e "$1" | lolcat
    else
        echo -e "$1"
    fi
}
```

```
show_header() {
    clear
    local ascii_art="
    .----- .
    /   o   /|  /\ \   \ \
    /_____/o|  /o \ \ o \ \
    |   o   ||  /  o \ \_____\ \
    |   o   |o/ \ \o  /o  /
    |   o   | /  \ \o / o  /
    |_____|  \ \ /____ /
    '-----'
  "
```

```

  A R C H   F O U N D R Y   &   A R M O R Y
  Repository Fabrication Unit - Mk. III
  "
```

```

# If lolcat exists, rainbow it. If not, use Steampunk
Bronze/Copper manually.
if command -v lolcat &> /dev/null; then
    echo "$ascii_art" | lolcat
else
    echo -e "${STEAM_BRONZE}"
    echo "
    .-----."
    echo "  /   o   /|   /\ \   \\"
    echo " /_____/o|   /o \ \ o  \\"
    echo " |   o   | |   /  o \ \___\\"
    echo " |   o   |o/  \ \o /o   /"
    echo " |   o   | /   \ \ o/ o  /"
    echo " '-----'   \ \___ /"
    echo -e "${NC}"
    echo -e "${STEAM_COPPER}"
}

echo -e "${STEAM_COPPER} ||   ${STEAM_GOLD}           A R C H   F O
U N D R Y   &   A R M O R Y   ${STEAM_COPPER}           ||${NC}"
echo -e "${STEAM_COPPER} ||   ${STEAM_IRON}           Repository
Fabrication Unit - Mk. III           ${STEAM_COPPER}           ||${NC}"
echo -e "${STEAM_COPPER}"

}

{NC}"
fi
echo
}

check_dependencies() {
    local deps=("git" "base-devel" "pacman-contrib")
    local missing=()
    for dep in "${deps[@]"; do
        if ! pacman -Qi "$dep" &> /dev/null; then
            missing+=("$dep")
        fi
    done

    if [ ${#missing[@]} -gt 0 ]; then
        echo -e "${STEAM_RUST}>> Stoking furnaces... Installing
missing components: ${missing[*]}${NC}"
        sudo pacman -S --noconfirm --needed "${missing[@]}"
    fi

    if ! command -v yay &> /dev/null; then
        echo -e "${STEAM_RUST}CRITICAL MALFUNCTION: 'yay' mechanism
not found.${NC}"
        exit 1
    fi
}

```

```

}

# --- Build Functions ---

setup_environment() {
    echo -e "${STEAM_BRONZE}>> Calibrating workspace
coordinates...${NC}"
    mkdir -p "$REPO_DIR"
    mkdir -p "$BUILD_DIR"
    rm -rf "$BUILD_DIR/*"
}

build_aur_packages() {
    echo -e "\n${STEAM_COPPER}>> Phase 1: Forging AUR Artifacts${NC}"
    echo -e "${STEAM_IRON}-----${NC}"

    if [ ! -f "$AUR_LIST" ]; then
        echo -e "${STEAM_RUST}Schematic not found at $AUR_LIST${NC}"
        return
    fi

    local current_dir=$(pwd)

    while read -r pkg; do
        [[ -z "$pkg" ]] && continue
        if ls "$REPO_DIR/$pkg"*.pkg.tar.zst 1> /dev/null 2>&1; then
            echo -e "${STEAM_GOLD}>> Artifact $pkg already in vault.
Skipping.${NC}"
            continue
        fi

        echo -e "${STEAM_BRONZE}>> Smelting: ${STEAM_WHITE}$pkg${NC}"
        cd "$BUILD_DIR" || exit

        if yay -G "$pkg" > /dev/null 2>&1; then
            cd "$pkg" || continue
            if makepkg -s --noconfirm --clean; then
                echo -e "${STEAM_GOLD}>> Forging complete: $pkg${NC}"
                mv *.pkg.tar.zst "$current_dir/$REPO_DIR/"
            else
                echo -e "${STEAM_RUST}>> Forging FAILED: $pkg${NC}"
            fi
        else
            echo -e "${STEAM_RUST}>> Could not acquire schematics for
$pkg.${NC}"
            fi
        cd "$current_dir" || exit
    done
}

```

```

done < "$AUR_LIST"
}

download_official_packages() {
    echo -e "\n${STEAM_COPPER}>> Phase 2: Acquiring Standard
Supplies${NC}"
    echo -e
"${STEAM_IRON}-----${NC}"

    if [ ! -f "$PKG_LIST" ]; then
        echo -e "${STEAM_RUST}Manifest not found at $PKG_LIST${NC}"
        return
    fi

    mapfile -t pkgs < "$PKG_LIST"
    local valid_pkgs=()
    for p in "${pkgs[@]"; do
        [[ -n "$p" ]] && valid_pkgs+=("$p")
    done

    echo -e "${STEAM_WHITE}>> Routing ${#valid_pkgs[@]} packages to
containment unit...${NC}"
    sudo pacman -Sw --cachedir "$REPO_DIR" --noconfirm --needed
"${valid_pkgs[@]}"
}

create_repository_db() {
    echo -e "\n${STEAM_COPPER}>> Phase 3: Engraving Repository
Database${NC}"
    echo -e
"${STEAM_IRON}-----${NC}"

    cd "$REPO_DIR" || exit
    local pkg_files=(*.pkg.tar.zst)

    if [ ${#pkg_files[@]} -eq 0 ]; then
        echo -e "${STEAM_RUST}No artifacts detected in the
vault.${NC}"
        return
    fi

    echo -e "${STEAM_BRONZE}>> Indexing ${#pkg_files[@]} items into
'$REPO_NAME.db.tar.gz'...${NC}"
    repo-add -n -R "$REPO_NAME.db.tar.gz" *.pkg.tar.zst
    cd ..
}

show_instructions() {

```

```

local abs_path=$(realpath "$REPO_DIR")
local footer="

    F A B R I C A T I O N   C O M P L E T E

Vault Location: $abs_path

To engage this machinery, modify /etc/pacman.conf:

[$REPO_NAME]
SigLevel = Optional TrustAll
Server = file://$abs_path

"

echo
if command -v lolcat &> /dev/null; then
    echo "$footer" | lolcat
else
    echo -e "${STEAM_GEAR}

    F A B R
I C A T I O N   C O M P L E T E    ${STEAM_GEAR}    ||${NC}"
    echo -e "${STEAM_GEAR}

    Vault Location:
${STEAM_WHITE}$abs_path ${STEAM_GEAR}    ||${NC}"
    echo -e "${STEAM_GEAR} ||
||${NC}"
    echo -e "${STEAM_GEAR} || ${STEAM_COPPER} To engage this
machinery, modify ${STEAM_WHITE}/etc/pacman.conf${STEAM_COPPER}:
${STEAM_GEAR}||${NC}"
    echo -e "${STEAM_GEAR} ||
||${NC}"
    echo -e "${STEAM_GEAR} || ${STEAM_WHITE} [$REPO_NAME]
${STEAM_GEAR}||${NC}"
    echo -e "${STEAM_GEAR} || ${STEAM_WHITE} SigLevel = Optional
TrustAll    ${STEAM_GEAR}||${NC}"
    echo -e "${STEAM_GEAR} || ${STEAM_WHITE} Server =
file://$abs_path    ${STEAM_GEAR}||${NC}"
    echo -e "${STEAM_GEAR}

    fi
    echo
}

```

```
# --- Main ---
show_header

if [ "$EUID" -eq 0 ]; then
    echo -e "${STEAM_RUST}CAUTION: Root access detected.${NC}"
    echo -e "${STEAM_WHITE}Protocol violation. Please operate this
machinery as a standard user.${NC}"
    exit 1
fi

check_dependencies
setup_environment
build_aur_packages
download_official_packages
create_repository_db
show_instructions
```