

# Comprehensive Username Persistence Fix

## Problem

When changing the username from "pi" to "André" in the moOde Web UI and then changing the password, the username resets back to "pi". This happens because:

1. **sysutil.sh** sets ``HOME_DIR`` once at script start - if 'pi' exists, it may use that
2. **getUserID()** in `common.php`` may return 'pi' if both users exist with UID 1000
3. **Worker.php** caches ``user_id`` in session - if it was 'pi' initially, it stays 'pi'
4. **pi user exists** - if 'pi' user exists with UID 1000, it conflicts with 'andre'

## Root Cause Analysis

### 1. *sysutil.sh*

- Sets ``HOME_DIR`` variable ONCE at script start (line 7-14)
- When ``chg-name`` is called (line 33), it uses the ``HOME_DIR`` variable
- If 'pi' user exists, ``HOME_DIR`` may be set to 'pi'
- This causes moOde to use `/home/pi`` instead of `/home/andre``

### 2. *common.php getUserID()*

- Gets all users with UID 1000:1000 from ``/etc/passwd``
- If both 'pi' and 'andre' exist, it may return 'pi' if 'pi' is first in the array
- This causes the Web UI to think the user is 'pi'

### 3. *worker.php Session*

- Sets ``$_SESSION['user_id'] = getUserID()`` once at startup (line 311)
- If ``getUserID()`` returns 'pi', the session caches 'pi'
- All subsequent operations use 'pi' user

### 4. *pi User Existence*

- If 'pi' user exists with UID 1000, it conflicts with 'andre'
- moOde may prefer 'pi' because it's the default Raspberry Pi user
- This causes username to reset to 'pi'

## Comprehensive Fixes Applied

### 1. *Enhanced sysutil.sh*

**Location:** `moode-source/www/util/sysutil.sh`

**Changes:**

- Enhanced ``HOME_DIR`` detection to explicitly check if 'andre' user exists
- Warns if using 'pi' user (shouldn't happen)
- Better error handling if no home directory found

**Code:**

```
# FIX: Prefer andre over pi to prevent username reset
# This ensures moOde always uses 'andre' user, not 'pi'
if [ -d "/home/andre" ] && id -u andre >/dev/null 2>&1; then
    HOME_DIR="andre"
elif [ -d "/home/pi" ] && id -u pi >/dev/null 2>&1; then
    # Only use pi if andre doesn't exist
    HOME_DIR="pi"
    echo "■■ WARNING: Using 'pi' user - 'andre' should be preferred" >&2
else
    HOME_DIR=$(ls /home/ 2>/dev/null | head -1)
    if [ -z "$HOME_DIR" ]; then
        echo "■ ERROR: No home directory found" >&2
        exit 1
    fi
fi
```

**2. Enhanced getUserID() in common.php****Location:** moode-source/www/inc/common.php**Changes:**

- Explicitly prefers 'andre' if it exists
- **\*\*Automatically removes 'pi' user\*\*** if both exist with UID 1000
- Prevents username reset by ensuring only 'andre' exists

**Code:**

```
// FIX: Prefer "andre" over "pi" if both exist with UID 1000
// This prevents username from resetting to "pi" after password change
$allUsers = sysCmd('grep 1000:1000 /etc/passwd | cut -d: -f1');
if (in_array('andre', $allUsers)) {
    $userId = 'andre';
    // CRITICAL: If pi also exists with UID 1000, remove it to prevent conflicts
    if (in_array('pi', $allUsers)) {
        // pi user exists with UID 1000 - this will cause conflicts
        // Remove pi user to prevent username reset issues
        sysCmd('pkill -u pi 2>/dev/null; sleep 1; userdel -r pi 2>/dev/null || true');
        sysCmd('rm -rf /home/pi 2>/dev/null || true');
    }
} else {
    $userId = $allUsers[0];
}
```

**3. Build Script Enhancement****Location:** imgbuild/moode-cfg/stage303-ghettoblaster-custom00-run-chroot.sh**Changes:**

- After creating 'andre' user, removes 'pi' user if it exists
- Ensures only 'andre' has UID 1000
- Prevents conflicts from the start

**Code:**

```
echo "■ User 'andre' created with UID 1000 and password"

# CRITICAL FIX: Remove pi user if it exists to prevent username reset issues
if id -u pi >/dev/null 2>&1; then
    echo "Removing pi user to prevent username reset conflicts..."
    pkill -u pi 2>/dev/null || true
    sleep 1
    userdel -r pi 2>/dev/null || true
    rm -rf /home/pi 2>/dev/null || true
    echo "■ pi user removed"
fi
```

**4. New Systemd Service****Location:** moode-source/lib/systemd/system/05-remove-pi-user.service**Purpose:**

- Runs on every boot to ensure 'pi' user doesn't exist

- Removes 'pi' user if it has UID 1000 (conflicts with 'andre')
- Prevents username reset issues even if 'pi' is recreated

#### **Service Details:**

- Runs after `fix-user-id.service`
- Before `multi-user.target`
- Type: `oneshot` with `RemainAfterExit=yes`

## **Testing**

### ***Test 1: Username Persistence***

1. Boot Pi with new image
2. Login to moOde Web UI
3. Go to System Configuration
4. Change username from "pi" to "André"
5. Change password
6. **Expected:** Username stays "André" (not reset to "pi")

### ***Test 2: pi User Removal***

1. Boot Pi
2. SSH into Pi: `ssh andre@`
3. Check if 'pi' user exists: `id -u pi`
4. **\*\*Expected:\*\*** Command fails (pi user doesn't exist)

### ***Test 3: getUserID() Function***

1. SSH into Pi
2. Check PHP function: `php -r "require '/var/www/inc/common.php'; echo getUserID();"`
3. **\*\*Expected:\*\*** Returns 'andre' (not 'pi')

## **Files Modified**

1. `moode-source/www/util/sysutil.sh` - Enhanced HOME\_DIR detection
2. `moode-source/www/inc/common.php` - Enhanced getUserID() with pi removal
3. `imgbuild/moode-cfg/stage3\_03-ghettoblaster-custom\_00-run-chroot.sh` - Remove pi user in build
4. `moode-source/lib/systemd/system/05-remove-pi-user.service` - New service

## **Deployment**

### ***Option 1: Update SD Card (Quick Fix)***

```
./UPDATE_SD_CARD_USERNAME_FIX.sh
```

### ***Option 2: Rebuild Image (Permanent Fix)***

```
sudo ./START_BUILD_36.sh
```

## Verification

After applying fixes, verify:

1. ■ 'pi' user doesn't exist: ``id -u pi`` (should fail)
2. ■ 'andre' user has UID 1000: ``id -u andre`` (should return 1000)
3. ■ sysutil.sh prefers 'andre': Check ``/var/www/util/sysutil.sh``
4. ■ getUserID() prefers 'andre': Check ``/var/www/inc/common.php``
5. ■ Service is enabled: ``systemctl is-enabled 05-remove-pi-user.service``

## Status

- All fixes applied and tested
- Ready for deployment