Channel Presets Enhancement Documentation

Overview

This enhancement adds comprehensive features to the channel preset system including:

- ✅ **Input Form** - Easy creation and editing of presets with name and channel number
- ✅ **Usage Tracking** - Automatic tracking of how often each preset is used
- ✅ **Data Backup** - Integrated backup system that preserves presets during updates
- ✅ **AI Auto-Reordering** - Intelligent reordering based on usage patterns
- ✅ **Persistent Data** - Presets survive GitHub updates

Features

1. Enhanced Input Form

The `ChannelPresetsPanel` component now includes:

- **Add Preset Form**: Enter friendly name (e.g., “ESPN”) and channel number (e.g., “27”)
- **Edit Preset**: Click edit icon to modify existing presets
- **Validation**: Channel numbers must be numeric
- **Device Type Tabs**: Separate presets for Cable Box and DirecTV
- **Manual Reordering**: Drag presets up/down to change order
- **Delete**: Remove unwanted presets

2. Usage Tracking

Database Fields Added:

- `usageCount` (Int): Tracks total number of times preset has been used
- `lastUsed` (DateTime): Records when preset was last used

Automatic Tracking:

- When a preset is used via the tune API, usage is automatically recorded
- Usage statistics are displayed on each preset card
- Shows “Used X times • Last: [date]”

API Endpoint:

```
POST /api/channel-presets/tune
Body: {
  channelNumber: "206",
  deviceType: "cable",
  presetId: "preset_id_here"  // Include this to track usage
}
```

3. Data Backup System

Backup Script: `scripts/backup-channel-presets.sh`

- Exports all active presets to JSON
- Stores in `~/sports-bar-backups/channel-presets/`

- Keeps last 30 backups
- Integrated into `update_from_github.sh`

Restore Script: `scripts/restore-channel-presets.sh`
- Automatically restores presets after GitHub updates
- Finds most recent backup
- Preserves usage statistics

Integration:
The `update_from_github.sh` script now:
1. Backs up channel presets before pulling updates
2. Pulls latest code from GitHub
3. Automatically restores presets from backup
4. Preserves all usage data

4. AI Auto-Reordering Service

Service: `src/services/presetReorderService.ts`

Features:
- Calculates weighted usage scores
- Recent usage weighted more heavily than old usage
- Time decay: usage loses value over time (50% every 30 days)
- Boost for very recent usage (within 24 hours)
- Separate ordering for cable and directv

Manual Trigger:
- Click "Auto-Reorder" button in the UI
- Or call API: `POST /api/channel-presets/reorder`

Automatic Scheduling:
Set up a cron job to run daily:

```
# Add to crontab (crontab -e)
0 3 * * * cd /home/ubuntu/Sports-Bar-TV-Controller && node scripts/reorder-presets-cron.ts >> logs/preset-reorder.log 2>&1
```

5. Usage Statistics API

Endpoint: `GET /api/channel-presets/statistics`

Returns:
- Statistics by device type (count, total usage)
- Top 10 most-used presets
- Useful for analytics and monitoring

API Reference

Get Presets

```
GET /api/channel-presets?deviceType=cable
Response: {
  success: true,
  presets: [...]
}
```

Create Preset

```
POST /api/channel-presets
Body: {
  name: "ESPN",
  channelNumber: "206",
  deviceType: "cable"
}
```

Update Preset

```
PUT /api/channel-presets/[id]
Body: {
  name: "ESPN HD",
  channelNumber: "207"
}
```

Delete Preset

```
DELETE /api/channel-presets/[id]
```

Tune to Channel (with usage tracking)

```
POST /api/channel-presets/tune
Body: {
  channelNumber: "206",
  deviceType: "cable",
  deviceIp: "192.168.1.100",  // Required for DirecTV
  presetId: "preset_id"       // Include to track usage
}
```

Trigger Auto-Reorder

```
POST /api/channel-presets/reorder
Response: {
  success: true,
  message: "Presets reordered successfully"
}
```

Get Usage Statistics

```
GET /api/channel-presets/statistics
Response: {
  success: true,
  statistics: [...],
  topPresets: [...]
}
```

Database Schema

```
model ChannelPreset {
  id            String    @id @default(cuid())
  name          String
  channelNumber String
  deviceType    String    // "cable" or "directv"
  order         Int       @default(0)
  isActive      Boolean   @default(true)
  usageCount    Int       @default(0)      // NEW
  lastUsed      DateTime?                  // NEW
  createdAt     DateTime  @default(now())
  updatedAt     DateTime  @updatedAt

  @@index([deviceType, order])
  @@index([isActive])
  @@index([usageCount])                    // NEW
}
```

Usage Scoring Algorithm

The AI reordering service uses a sophisticated scoring algorithm:

```
Base Score = usageCount × 100

Time Decay = 0.5 ^ (daysSinceLastUse / 30)
Score = Base Score × Time Decay

Boosts:
- Recent usage (< 24 hours): 1.5x multiplier
- New presets (< 7 days old): 1.1x multiplier
```

This ensures:
- Frequently used presets stay at the top
- Old unused presets gradually move down
- Recently used presets get priority
- New presets get a fair chance

Backup and Restore

Manual Backup

```
cd /home/ubuntu/Sports-Bar-TV-Controller
./scripts/backup-channel-presets.sh
```

Manual Restore

```
cd /home/ubuntu/Sports-Bar-TV-Controller
./scripts/restore-channel-presets.sh
```

Automatic Backup (during updates)

The backup happens automatically when running:

```
./update_from_github.sh
```

Testing

Test the UI

1. Navigate to Settings → Channel Presets
2. Add a new preset (e.g., “ESPN”, channel “206”)
3. Edit the preset
4. Use the preset (tune to channel)
5. Check that usage count increments
6. Click “Auto-Reorder” to test reordering

Test the API

```
# Create a preset
curl -X POST http://localhost:3000/api/channel-presets \
  -H "Content-Type: application/json" \
  -d '{"name":"ESPN","channelNumber":"206","deviceType":"cable"}'

# Get presets
curl http://localhost:3000/api/channel-presets?deviceType=cable

# Trigger reorder
curl -X POST http://localhost:3000/api/channel-presets/reorder

# Get statistics
curl http://localhost:3000/api/channel-presets/statistics
```

Test Backup/Restore

```
# Create some test presets
# Run backup
./scripts/backup-channel-presets.sh

# Verify backup file exists
ls -la ~/sports-bar-backups/channel-presets/

# Test restore
./scripts/restore-channel-presets.sh
```

Migration Notes

The enhancement includes a database migration that adds:
- `usageCount` field (default: 0)
- `lastUsed` field (nullable)
- Index on `usageCount` for efficient sorting

Migration is automatically applied when running:

```
npx prisma migrate deploy
```

Files Modified/Created

New Files

- `src/services/presetReorderService.ts` - AI reordering logic
- `src/app/api/channel-presets/reorder/route.ts` - Manual reorder API
- `src/app/api/channel-presets/statistics/route.ts` - Statistics API
- `scripts/backup-channel-presets.sh` - Backup script
- `scripts/restore-channel-presets.sh` - Restore script
- `scripts/reorder-presets-cron.ts` - Cron job script
- `prisma/migrations/20250103_add_usage_tracking/migration.sql` - Database migration

Modified Files

- `prisma/schema.prisma` - Added usage tracking fields
- `src/app/api/channel-presets/tune/route.ts` - Added usage tracking
- `src/components/settings/ChannelPresetsPanel.tsx` - Enhanced UI
- `update_from_github.sh` - Integrated backup/restore

Future Enhancements

Potential future improvements:
- Export/import presets as JSON
- Share presets between users
- Preset categories/tags
- Search and filter presets
- Bulk operations
- Usage analytics dashboard

- Machine learning for better predictions
- Integration with TV guide data

Troubleshooting

Presets not loading

- Check database exists: `ls -la prisma/data/sports_bar.db`
- Check migrations applied: `npx prisma migrate status`
- Check API logs for errors

Usage not tracking

- Ensure `presetId` is included in tune API calls
- Check browser console for errors
- Verify database has `usageCount` and `lastUsed` fields

Backup/restore issues

- Check backup directory exists: `~/sports-bar-backups/channel-presets/`
- Verify scripts are executable: `chmod +x scripts/*.sh`
- Check database path in scripts

Auto-reorder not working

- Check service logs for errors
- Verify presets have usage data
- Test manually via API first

Support

For issues or questions:
1. Check the logs: `tail -f logs/preset-reorder.log`
2. Review API responses in browser console
3. Check database state: `sqlite3 prisma/data/sports_bar.db "SELECT * FROM ChannelPreset;"`
4. Verify migrations: `npx prisma migrate status`