**📚 MIKE-GEN-LIBS USER GUIDE**

**Complete Workflows for Rapid Composition**

**🎯 QUICK START CHECKLIST**

* [ ] mike-gen-libs installed at C:\Users\mike\Documents\gml-workspace\mike-gen-libs
* [ ] Export system files in place (src/export/universal\_export.js)
* [ ] Rapid Quintet Generator ready (rapid-quintet-generator.html)
* [ ] Your GML apps accessible (RiffGen, TriadGen, Quartet Engine, etc.)

**🎻 WORKFLOW 1: RAPID STRING QUINTET (3 Minutes)**

**Step-by-Step:**

1. **Open Rapid Quintet Generator**
2. cd /c/Users/mike/Documents/gml-workspace/mike-gen-libs
3. explorer rapid-quintet-generator.html
4. **Generate Base Quintet**
   * Click GENERATE QUINTET
   * Set Tempo: 120 BPM
   * Set Key: D Major
5. **Export to RiffGen for Melody Enhancement**
   * Click EXPORT OPTIONS
   * Click Export to RiffGen
   * RiffGen opens with Violin 1 melody
   * Add ornaments, trills, variations
6. **Return Enhanced Melody to Quintet**
   * In RiffGen: Click Export to Quintet
   * Quintet Composer receives enhanced melody
   * Auto-distributes to all 5 string parts

**Result:** Full string quintet in under 3 minutes!

**🎸 WORKFLOW 2: GUITAR + STRING QUARTET FUSION**

**The Dave Gilmour Section (Atmospheric Lead)**

**Step 1: Load Guitarist Profile**

// In GML-Guitar-Profiles-Library

loadProfile('David\_Gilmour\_Lead.json')

// Exports: Sustained bends, delay-heavy phrases

**Step 2: Generate Complementary Strings**

1. **Open TriadGen**
   * Generate Em - G - D - Am progression
   * Export to Quartet Engine
2. **In Quartet Engine**
   * Receive chord progression
   * Auto-arrange for strings:
     + Violin 1: Tremolo on high harmonics
     + Violin 2: Sustained counter-melody
     + Viola: Rhythmic pizzicato
     + Cello: Deep pedal tones
3. **Combine in GML-ACE**
4. // GML-ACE receives both:
5. guitarPart = importFrom('Guitar-Profiles')
6. stringQuartet = importFrom('Quartet-Engine')
7. // Auto-blend:
8. finalArrangement = blend(guitarPart, stringQuartet, {
9. guitarVolume: 0.7,
10. stringsVolume: 0.5,
11. reverb: 'hall'
12. })

**Result:** Pink Floyd-style atmospheric section with guitar floating over strings

**🎼 WORKFLOW 3: MULTI-COMPOSER COLLABORATION**

**Section A: Classical Opening (Mozart Profile)**

1. **In Composer Profiles Extension**
2. loadComposer('Mozart\_Chamber.json')
3. generateSection({
4. instruments: ['violin1', 'violin2', 'viola', 'cello'],
5. measures: 16,
6. style: 'classical\_sonata'
7. })
8. **Export to Quintet Composer**
   * Add 5th part (Cello 2) as bass double
   * Export checkpoint: mozart\_section\_A.json

**Section B: Jazz Fusion Bridge (Pat Metheny Profile)**

1. **In Guitar Profiles**
2. loadProfile('Pat\_Metheny\_Jazz.json')
3. generateBridge({
4. chords: ['Gmaj7', 'Em9', 'Am11', 'D13'],
5. measures: 8
6. })
7. **Send to String FX Mapper**
   * Apply: Jazz articulations
   * Techniques: Slides, ghost notes, swing feel
   * Export to Quintet

**Section C: Rock Finale (Jimmy Page Profile)**

1. **Guitar Profiles → Generate Power Section**
2. loadProfile('Jimmy\_Page\_Riffs.json')
3. generateFinale({
4. intensity: 'high',
5. techniques: ['power\_chords', 'palm\_muting', 'harmonics']
6. })
7. **Infinite Riff Machine → Create Variations**
   * Take Page riff
   * Generate 4 variations
   * Export strongest to Quartet Engine
8. **Combine All Sections in GCC**
9. // In Generative Chamber Composer
10. finalPiece = {
11. sectionA: import('mozart\_section\_A.json'), // 0:00-1:30
12. sectionB: import('metheny\_bridge\_B.json'), // 1:30-2:15
13. sectionC: import('page\_finale\_C.json'), // 2:15-3:30
14. coda: generateTransition(sectionC, sectionA) // 3:30-4:00
15. }

**Result:** 4-minute piece blending Classical→Jazz→Rock with smooth transitions

**🚀 WORKFLOW 4: RAPID 10-PIECE ARRANGEMENT**

**Goal: Guitar Duo + String Quintet + Rhythm Section**

**Phase 1: Rhythm Foundation (30 seconds)**

1. **Drum Engine**
2. generateGroove({ style: 'fusion', tempo: 110, pattern: '16th\_note\_funk'})// Export to GML-ACE

**Phase 2: Harmonic Structure (45 seconds)**

1. **TriadGen → Quick Progression**
   * C - Am - F - G (4 bars each)
   * Export to both Quartet Engine AND Guitar Profiles

**Phase 3: String Arrangement (60 seconds)**

1. **Quartet Engine receives triads**
   * Auto-voices for 4 parts
   * Export to Quintet Composer
2. **Quintet Composer**
   * Adds 5th voice (Cello 2)
   * Creates contrary motion
   * Export to GML-ACE

**Phase 4: Guitar Layers (45 seconds)**

1. **Guitar Profile 1: Rhythm (John Mayer style)**
2. loadProfile('John\_Mayer\_Rhythm.json')
3. generateComping(chordProgression)
4. **Guitar Profile 2: Lead (Eric Clapton style)**
5. loadProfile('Clapton\_Blues\_Lead.json')
6. generateSolo({
7. scale: 'C\_pentatonic',
8. intensity: 'medium'
9. })

**Phase 5: Final Assembly (30 seconds)**

1. **GML-ACE: Combine Everything**
2. masterArrangement = { drums: importFrom('Drum-Engine'), bass: generateBassFrom(chordProgression), guitarRhythm: importFrom('Guitar-Profile-1'), guitarLead: importFrom('Guitar-Profile-2'), strings: importFrom('Quintet-Composer')}// Auto-mix with proper levelsfinalMix = autoBalance(masterArrangement)exportTo('MIDI\_and\_MusicXML')

**Total Time: ~3 minutes** **Result:** Professional 10-piece arrangement

**💡 WORKFLOW 5: THEME & VARIATIONS**

**Original Theme Generation**

1. **RiffGen: Create 8-bar melody**
2. theme = generateRiff({ length: 8, key: 'G\_major', contour: 'arch'})

**Variation 1: String Quintet Treatment**

1. **Export theme to Quintet Composer**
2. **Auto-harmonize for 5 voices**
3. **Apply baroque counterpoint rules**

**Variation 2: Guitar Hero Version**

1. **Send theme to Guitar Profiles**
2. **Load 'Steve\_Vai\_Virtuoso.json'**
3. **Add: Tapping, sweeps, harmonics**

**Variation 3: Jazz Reharmonization**

1. **Theme → TriadGen**
2. **Convert to jazz changes:**
   * Original: G - C - D
   * Jazz: Gmaj7 - Cm9 - D7alt

**Variation 4: Orchestral Finale**

1. **All variations → GCC**
2. **Layer and combine:**
   * Strings play original theme
   * Guitar plays virtuoso variation
   * Full ensemble on jazz changes

**🎯 WORKFLOW 6: QUICK FILM SCORE**

**Scene: Chase Sequence (90 seconds needed)**

**0:00-0:30 - Building Tension**

1. **Quintet Composer**
   * Generate tremolo strings in minor key
   * Export to GML-ACE

**0:30-1:00 - Action Peaks**

1. **Drum Engine + Guitar Profiles**
2. // DrumsgeneratePattern('action\_chase\_drums')// Guitar (Metal profile)loadProfile('Metallica\_Rhythm.json')generateRiff('palm\_muted\_sixteenths')

**1:00-1:30 - Resolution**

1. **TriadGen → Quartet Engine**
   * Major resolution chords
   * Quartet voices in contrary motion
   * Add Guitar Profile: 'Ambient\_Clean.json'

**Export complete score in 5 minutes!**

**📊 ADVANCED WORKFLOW: ALGORITHMIC COMPOSITION**

**Using Multiple Profiles Simultaneously**

// In GML-ACE or custom script

function generateMultiStylePiece() {

// Load 3 guitarist profiles

const guitarists = [

loadProfile('Hendrix\_Psychedelic.json'),

loadProfile('Segovia\_Classical.json'),

loadProfile('Pass\_Jazz.json')

];

// Load 3 composer profiles

const composers = [

loadComposer('Bach\_Fugue.json'),

loadComposer('Debussy\_Impressionist.json'),

loadComposer('Glass\_Minimalist.json')

];

// Generate sections

sections = [];

// Each guitarist plays with each composer

guitarists.forEach((guitarist, g) => {

composers.forEach((composer, c) => {

const section = {

guitar: guitarist.generate(8), // 8 bars

strings: composer.generate({

ensemble: 'string\_quintet',

measures: 8,

harmonicLanguage: composer.style

})

};

// Export each combination

exportToApp({

sectionId: `G${g}\_C${c}`,

data: section

}, 'QuintetComposer');

sections.push(section);

});

});

// Create final 9-section piece (3x3 grid)

return assembleFinalPiece(sections);

}

**Result:** 9 unique style combinations in one piece!

**🔥 SPEED COMPOSITION CHALLENGES**

**Challenge 1: "2-Minute Symphony"**

1. 0:00-0:30 - Generate quintet (Quintet Composer)
2. 0:30-1:00 - Add guitar lead (Guitar Profiles)
3. 1:00-1:30 - Create variations (Infinite Riff Machine)
4. 1:30-2:00 - Mix and export (GML-ACE)

**Challenge 2: "Style Morph"**

* Start with Mozart-style string quartet
* Every 30 seconds, blend in new style:
  + 0:30 - Add blues guitar
  + 1:00 - Shift to jazz harmony
  + 1:30 - End with metal power chords
* Use smooth transitions via GCC

**Challenge 3: "The Relay"**

* RiffGen creates 4-bar melody (20 sec)
* TriadGen harmonizes it (20 sec)
* Quartet Engine arranges it (20 sec)
* Guitar Profiles adds solo (20 sec)
* Quintet Composer adds 5th voice (20 sec)
* GML-ACE final mix (20 sec)
* **Total: 2 minutes for complete piece!**

**📝 EXPORT FORMATS REFERENCE**

**From Any App to Any App:**

**Small Data (< 2KB) - URL Parameters**

// Automatic for small transfers

exportToApp(smallData, 'TargetApp')

// Creates: http://localhost:3004?import=...

**Large Data (> 2KB) - LocalStorage**

// Automatic for large transfers

exportToApp(largeData, 'TargetApp')

// Creates: localStorage key + URL reference

**Export Chains**

RiffGen → Quintet → GML-ACE → Final Output

TriadGen → Quartet → Quintet → Final Output

Guitar → String FX → Quartet → Final Output

Drum → GCC → GML-ACE → Final Output

**🛠️ TROUBLESHOOTING**

**Issue: "Export button does nothing"**

**Solution:** Check console for errors, verify universal\_export.js is loaded

**Issue: "Can't find my exported data"**

**Solution:** Check localStorage: localStorage.getItem('gml\_export\_[timestamp]')

**Issue: "Apps won't connect"**

**Solution:** Ensure all apps reference same mike-gen-libs folder

**Issue: "Tempo doesn't match"**

**Solution:** Standardize in export: data.tempo = 120

**🚀 PRODUCTIVITY TIPS**

1. **Create Templates**
   * Save favorite progressions in TriadGen
   * Save quintet templates with your preferred voicings
   * Keep guitar profile combinations that work well
2. **Keyboard Shortcuts** (add to your apps)
   * Ctrl+G = Generate
   * Ctrl+E = Export
   * Ctrl+Q = Quick Quintet
3. **Batch Processing**
4. // Generate 10 variations at once
5. for(let i = 0; i < 10; i++) {
6. generateQuintet();
7. exportToApp(currentQuintet, 'RiffGen');
8. }
9. **Chain Reactions**
   * Set up automatic chains:
   * When RiffGen exports → Quintet auto-harmonizes
   * When Quintet exports → GML-ACE auto-mixes

**📊 COMPLETE APP NETWORK MAP**

┌─────────────┐ ┌──────────────┐ ┌─────────────┐

│ RiffGen │────▶│ Quintet │────▶│ GML-ACE │

└─────────────┘ └──────────────┘ └─────────────┘

│ ▲ ▲

▼ │ │

┌─────────────┐ ┌──────────────┐ │

│ TriadGen │────▶│ Quartet │────────────┘

└─────────────┘ └──────────────┘

│

▼

┌─────────────┐ ┌──────────────┐ ┌─────────────┐

│Guitar Prof. │────▶│ String FX │────▶│ GCC │

└─────────────┘ └──────────────┘ └─────────────┘

│

▼

┌──────────────┐ ┌─────────────┐

│ Infinite Riff│────▶│ Drum Engine │

└──────────────┘ └─────────────┘

**✨ REMEMBER:**

**Every app can talk to every other app through mike-gen-libs!**

**Your Universal Export System makes rapid composition not just possible, but effortless.**

*Last Updated: 2024* *Version: BULLETPROOF-9x3* *Location: C:\Users\mike\Documents\gml-workspace\mike-gen-libs\USER\_GUIDE.md*