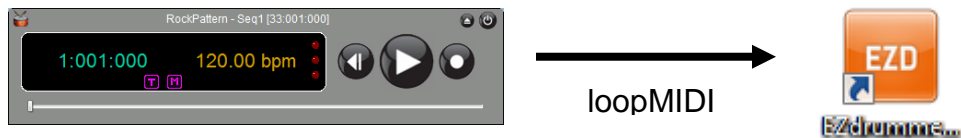


Drum Gen

Tutorial #2

Creating a Full Drum Track for EZ Drummer 2

In Tutorial #1 we configured DrumGen and EZ Drummer 2 to communicate using the virtual MIDI cable *loopMIDI*.



If you use drum software other than EZ Drummer 2, Tutorial #1 is still a guide to configure DrumGen to drive your software.

The examples in Tutorial #2 are specific to **EZ Drummer 2**. However, the workflow is similar when DrumGen is driving other drum software. We will use EZ Drummer 2's *Jazz Basic* drum kit to play back the drum track we create for a jazz song.



Using a Map File

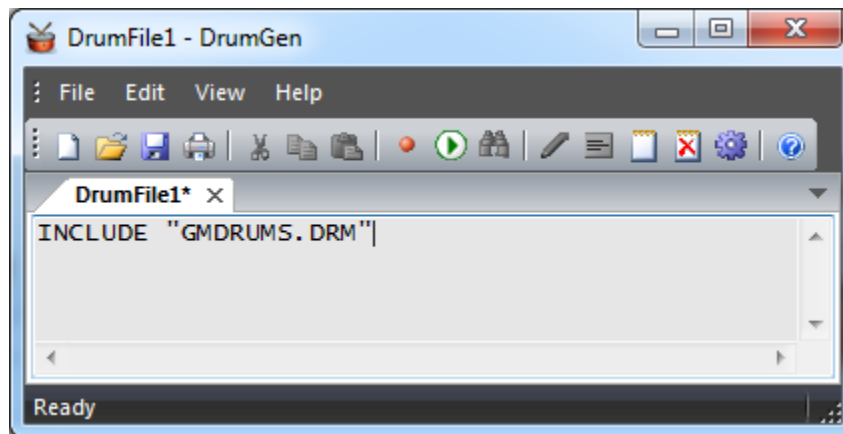
A *drum map* allows a MIDI drum pattern programmed using a certain note organization - e.g. the General MIDI (GM) standard note definitions - to be played back on a drum kit that might have a different note organization.

Most modern drum software - including EZ Drummer 2 - follows the GM standard (for the most part) and will play back the drum portion of GM-compliant MIDI files, whether they contain standalone drum loops, or full multi-instrumental songs with the drum track on MIDI channel 10.

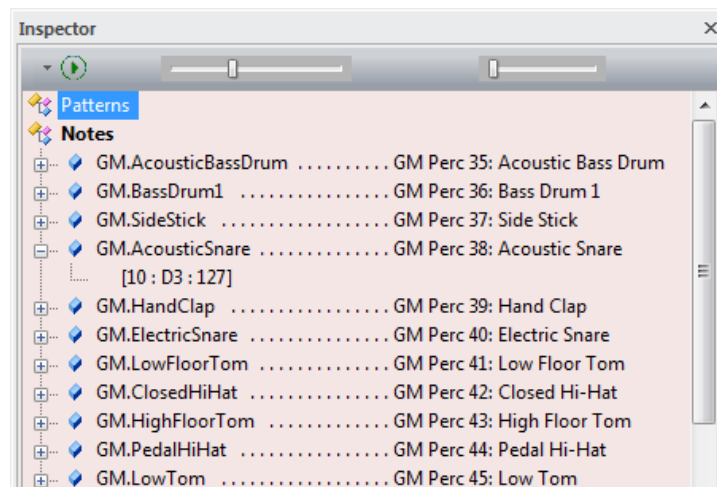
To play back simple DrumGen drum programs on EZ Drummer 2, we will start by using the *most basic of map files*, the one that creates General MIDI Note definitions:

```
INCLUDE "GMDRUMS.DRM"
```

Let's create a drum program with the single line above. (If you haven't yet configured DrumGen to play back on EZ Drummer 2, do so now using Tutorial #1).



When we compile the simple drum program above and then click into the Inspector tool we see a list of Note definitions:



Within the Inspector, if you click on the GM.AcousticSnare line item, you will hear a snare sound from EZ Drummer 2. **The EZ Drummer 2 Jazz Basic kit is GM-compatible for the most part.**

There I am saying "for the most part" again. Clicking on GM.AcousticSnare in the Inspector, or using that particular Note definition in a Pattern as below, will cause a [10:D3] MIDI message to be sent to EZ Drummer 2 (a 'D3' note on MIDI channel 10), per the GM standard.

```
PATTERN p {  
    ...  
    GM.AcousticSnare      4      120  
    ...  
}
```

Many GM drum tracks will play back as expected within the *Jazz Basic* kit. However it isn't exactly a GM drum kit because there are *missing notes and extra notes*. **This is the subject of Tutorial #3**; we will just stick to the basics for this tutorial.

Laying Out a Drum Track

We will create an uptempo drum track for a standard 32-bar A-A-B-A jazz song form using shuffle patterns. The Patterns we will need to insert into the various Sections are:

```
Intro_4  
Verse_8  
Bridge_8  
Verse_Solo_8  
Bridge_Solo_8  
Coda_4
```

We will start the Play Section with four bars of Introduction - a light timekeeping Pattern:

```
PLAY  
  
SECTION "Intro"  
    Intro_4
```

The first 32-bar chorus will be for the "head" - the jazz tune itself with light drums:

```
SECTION 32 "Head"  
    Verse_8    Verse_8    Bridge_8    Verse_8
```

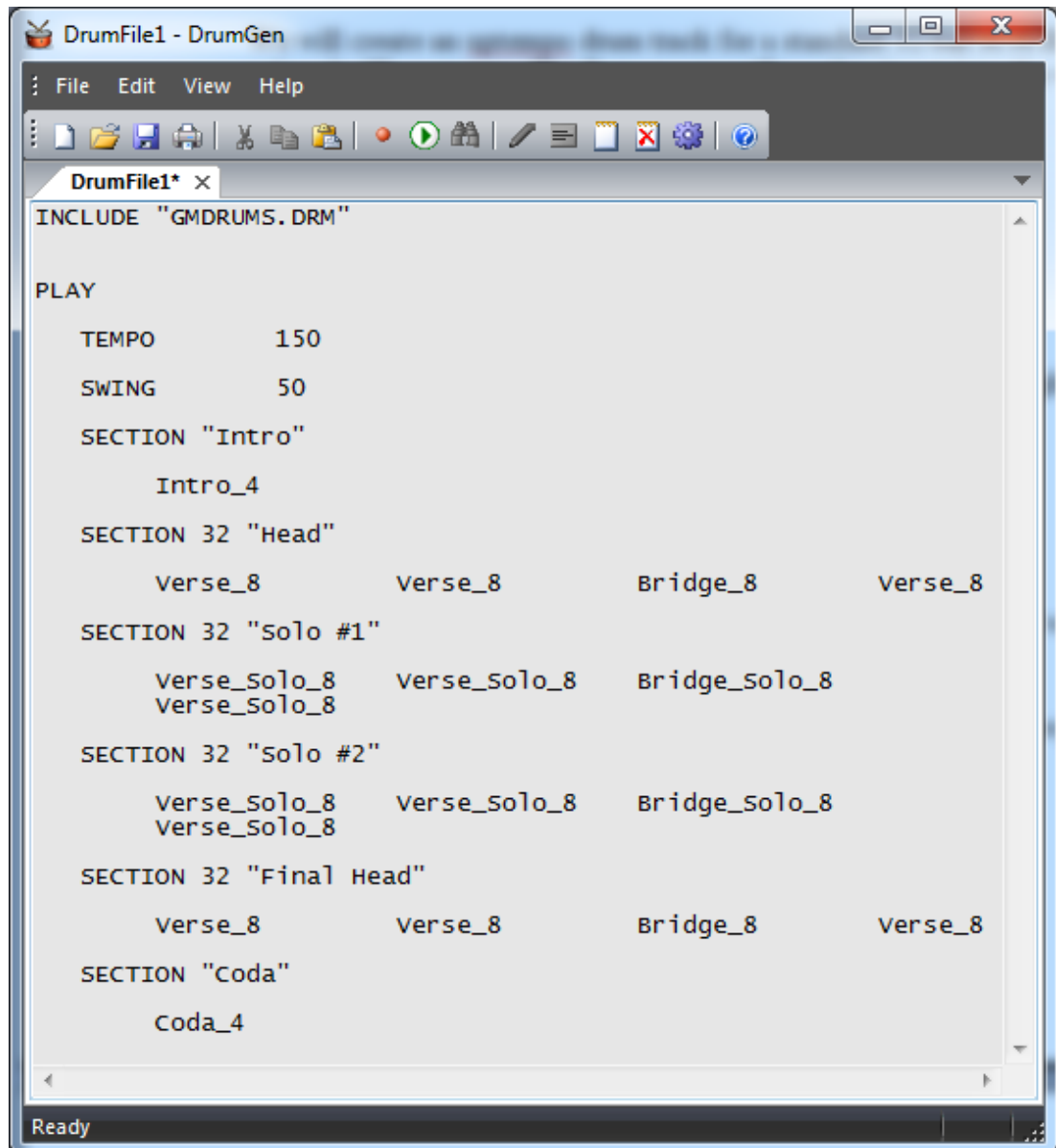
Subsequent 32-bar choruses will be for instrumental soloing and will use a more active drum accompaniment, thus different Patterns:

```
SECTION 32 "Solo #1"  
  Verse_Solo_8  Verse_ Solo_8  Bridge_Solo_8  
  Verse_Solo_8
```

After returning to the head once again after solos, the song will end with a short coda:

```
SECTION "Coda"  
  Coda_4
```

Our drum program structure is now:



The screenshot shows a window titled "DrumFile1 - DrumGen" with a menu bar (File, Edit, View, Help) and a toolbar. The main text area contains the following drum program structure:

```
DrumFile1* x  
INCLUDE "GMDRUMS.DRM"  
  
PLAY  
  TEMPO      150  
  SWING      50  
  SECTION "Intro"  
    Intro_4  
  SECTION 32 "Head"  
    Verse_8      Verse_8      Bridge_8      Verse_8  
  SECTION 32 "Solo #1"  
    Verse_Solo_8 Verse_Solo_8 Bridge_Solo_8  
    Verse_Solo_8  
  SECTION 32 "Solo #2"  
    Verse_Solo_8 Verse_Solo_8 Bridge_Solo_8  
    Verse_Solo_8  
  SECTION 32 "Final Head"  
    Verse_8      Verse_8      Bridge_8      Verse_8  
  SECTION "Coda"  
    Coda_4
```

The status bar at the bottom left of the window displays "Ready".

Defining Patterns

We will create reusable 4-bar and 8-bar Patterns in a separate drum program file `Tut2_Include.drm` which we will include in our main drum program (now called `Tutorial2.drm`). These Pattern definitions, named within the context "Tut2", will now be available in our main drum program and we will use them to fill in the Patterns we've named above in our Sections, i.e. *we will use them as subpatterns*, for example:

```
PATTERN Intro_4      { /Tut2.Intro }
```

The Definition Section is now expanded to be:

```
// *****
// Tutorial #2
// *****

NOLISTING
  INCLUDE "GMDRUMS.DRM"
LISTING

INCLUDEDIR "D:\DrumGen Tutorials"

INCLUDE "Tut2_Include.drm"

PATTERN Intro_4      { /Tut2.Intro }
PATTERN Verse_8      { TEXT "Verse_8"      /Tut2.Verse }
PATTERN Bridge_8     { TEXT "Bridge_8"     /Tut2.Verse }
PATTERN Verse_Solo_8 { TEXT "Verse_Solo_8" /Tut2.Verse_Alt }
PATTERN Bridge_Solo_8 { TEXT "Bridge_Solo_8" /Tut2.Verse_Alt }
PATTERN Coda_4       { /Tut2.Coda }
```

I won't describe the details of the included Patterns (please refer to the entire drum program at the end of this tutorial), but I will highlight a few things. In most cases the Patterns are constructed with subpatterns "_R", "_L", "_RF", and "_LF" corresponding to the notes played by the hands and feet of a drummer. The Verse Pattern is repeated throughout the drum track and thus benefits from *variability* provided by constructs such as velocity ranges and the 'or' operator (|) which allows random selection. Examples within the Patterns are:

```
DEFINE RideVol      30..50
DEFINE SnareVol     30..80

NOTEDEF RideNote   [GM.RideBell | GM.AcousticSnare]

// -- Bar 4 .. occasional ruff

[ /SnareRuff_1 | { REST 1 } | { REST 1 } ]
```

The final drum track may now be played using EZ Drummer 2.

The screenshot displays the EZ Drummer 2 interface. At the top, the window title is 'Tutorial2.drm - STAFF - DrumGen'. The menu bar includes 'File', 'View', and 'Help'. Below the menu is a toolbar with various icons for editing and playback. The main workspace shows a piano roll for a drum track. The piano roll has a vertical axis labeled 'Head' with markers for measures 4, 5, 6, 7, and 8. The horizontal axis represents time. The piano roll contains several drum events, including Bass Drum, Snare, Low Tom, Closed HH, Open HH, LoMidTom, HiMidTom, Crash, and Ride/Bell. The track is labeled 'Verse_8' in the bottom right corner. The status bar at the bottom indicates 'Ready'.

Our final main drum program Tutorial2.drm is short since all the Pattern details are in the include file (see **bold** statement below). Feel free to cut-and-paste the text below.

```
// *****
// Tutorial #2
// *****

NOLISTING
  INCLUDE "GMDRUMS.DRM"
LISTING

INCLUDEDIR "D:\DrumGen Tutorials"

INCLUDE "Tut2_Include.drm"

PATTERN Intro_4      { /Tut2.Intro }
PATTERN Verse_8      { TEXT "Verse_8" /Tut2.Verse }
PATTERN Bridge_8     { TEXT "Bridge_8" /Tut2.Verse }
PATTERN Verse_Solo_8 { TEXT "Verse_Solo_8" /Tut2.Verse_Alt }
PATTERN Bridge_Solo_8 { TEXT "Bridge_Solo_8" /Tut2.Verse_Alt }
PATTERN Coda_4       { /Tut2.Coda }

PLAY

TEMPO 150
SWING 50

SECTION "Intro"

  Intro_4

SECTION 32 "Head"

  Verse_8      Verse_8      Bridge_8      Verse_8

SECTION 32 "Solo #1"

  Verse_Solo_8 Verse_Solo_8 Bridge_Solo_8
  Verse_Solo_8

SECTION 32 "Solo #2"

  Verse_Solo_8 Verse_Solo_8 Bridge_Solo_8
  Verse_Solo_8

SECTION 32 "Final Head"

  Verse_8      Verse_8      Bridge_8      Verse_8

SECTION "Coda"

  Coda_4
```

The include file Tut2_Include.drm (**bolding** added):

```
CONTEXT "Tut2"

// ==== INTRO, CODA =====

NOSHOW

PATTERN Intro_R {
    DESCRIPTION    "Intro - sticks"

// -- Bar 1-2, 3-4

    {
        GM.SideStick      4      70
        GM.SideStick      4      50    *3

        GM.SideStick      4      50
        GM.SideStick      4      50
        REST               8
        GM.SideStick      8      60
        GM.SideStick      4      55

    } *2
}

PATTERN Intro_L {
    DESCRIPTION    "Intro - toms"

    REST           1          *3

// -- Bar 4

    REST           4 dot dot TIE 32

    GM.LowFloorTom 32    50    // ruff
    GM.LowFloorTom 8     90
    GM.LowFloorTom 4     50
    GM.LowFloorTom 8     50

}

PATTERN Intro_RF {
    DESCRIPTION    "Intro - kick"

    REST           1          *2
    REST           2 dot dot TIE 16

// -- Bar 4

    | >20 |
    GM.BassDrum1   16     50    // ruff
```



```

GM.BassDrum1      4      90
}

SHOW

PATTERN Intro {
    [ /Intro_R + /Intro_L + /Intro_RF ]
}

PATTERN Coda {
// -- Bar 1

GM.BassDrum1      4 dot dot  90
GM.BassDrum1      16          50    // ruff
GM.BassDrum1      4 dot dot  90
REST              32

// -- Bar 2

GM.HiMidTom       32      50          // ruff
GM.HiMidTom       6       60..99      *3
GM.LowMidTom      6       60..99      *3

// -- Bar 3-4  fade with cymbal flare

GM.CrashCymbal1   :1      60

GM.CrashCymbal1   32      CRESC 0..60 *32
GM.CrashCymbal1   32      CRESC 60..0 *32
}

// ===== VERSE =====

NOSHOW

DEFINE RideVol     30..50
DEFINE SnareVol    30..80

NOTEDEF RideNote [GM.RideBell | GM.AcousticSnare]

PATTERN SnareRuff_1 {
    DESCRIPTION    "NOSHOW"
    REST           4 dot dot TIE 32
    GM.AcousticSnare 32      70    // ruff
    GM.AcousticSnare 2       70..90
}

PATTERN Verse_R {
    DESCRIPTION    "Verse - ride & snare"

```

```

// -- Bar 1
    REST                1

// -- Bar 2
    RideNote            4    RideVol
    REST                8
    RideNote            8    RideVol
    RideNote            4    RideVol
    REST                4

// -- Bar 3
    RideNote            4    RideVol
    REST                8
    RideNote            8    RideVol
    RideNote            4    RideVol
    REST                4

// -- Bar 4 .. occasional ruff
    [ /SnareRuff_1 | { REST 1 } | { REST 1 } ]

// -- Bar 5
    REST                1

// -- Bar 6
    RideNote            4    RideVol
    REST                8
    RideNote            8    RideVol
    RideNote            4    RideVol
    REST                4

// -- Bar 7
    RideNote            4    RideVol
    REST                8
    RideNote            8    RideVol
    RideNote            4    RideVol
    REST                4

// -- Bar 8
    REST                4 dot dot TIE 32
    GM.AcousticSnare    32    SnareVol    // ruff
    GM.AcousticSnare    8     SnareVol
    GM.AcousticSnare    4     SnareVol
    GM.AcousticSnare    8     SnareVol
}

```

PATTERN Verse_R_Alt {

```
DESCRIPTION      "Verse - ride & snare"

// -- Bar 1-3

{
  RideNote      4      RideVol
  REST          8
  RideNote      8      RideVol
  RideNote      4      RideVol
  REST          4
} *3

// -- Bar 4

[
  {
    REST          4 dot dot TIE 32
    GM.AcousticSnare 32 70 // ruff
    GM.AcousticSnare 2 70..90
  } | { REST 1 } | { REST 1 }
]

// -- Bar 5-7

{
  RideNote      4      RideVol
  REST          8
  RideNote      8      RideVol
  RideNote      4      RideVol
  REST          4
} *3

// -- Bar 8

REST          4 dot dot TIE 32
GM.AcousticSnare 32 SnareVol // ruff
GM.AcousticSnare 8 SnareVol
GM.AcousticSnare 4 SnareVol
GM.AcousticSnare 8 SnareVol
}
```

PATTERN Verse_L {

```
DESCRIPTION      "Verse - hi-hat"

{
  GM.OpenHiHat      4      60..90      *2
  GM.OpenHiHat      8      60..90
  GM.ClosedHiHat    8      60..90
  GM.OpenHiHat      4      60..90
} *8
}
```

```

PATTERN Verse_RF {
    DESCRIPTION      "Verse - kick"

    GM.BassDrum1      1      90

    REST              1 *2
    REST              2 dot dot TIE 16

    GM.BassDrum1     16     50    // ruff
    GM.BassDrum1     1      60

    REST              1 *3
}

PATTERN Verse_LF {
    DESCRIPTION      "Verse - hi-hat"

    {
        REST          4
        GM.PedalHiHat 4
    } *16
}

PATTERN Verse_Extra {
// -- Bar 1

    GM.CrashCymbal1  1      60
    REST              1 *3

// -- Bar 5

    GM.CrashCymbal1  1      60
    REST              1 *3
}

SHOW

PATTERN Verse {
    [ /Verse_R + /Verse_L + /Verse_RF + /Verse_LF + /Verse_Extra ]
}

PATTERN Verse_Alt {
    [ /Verse_R_Alt + /Verse_L + /Verse_RF + /Verse_LF +
      /Verse_Extra ]
}

ENDCONTEXT "Tut2"

```