V21.1.0 | 14th Jan 2021

IME 21 User Guide |

V21.1.0 | 14th Jan 2021 | ISE IWS Pak | Wotja

Reflective Generative MIDI Music

The **I**ntermorphic (MIDI) **M**usic **E**ngine (**IME**) 21 is an 'adaptive' generative music engine and is the latest iteration in our line of IME engines. It is also the evolution of the SSEYO Koan Generative Music Engine (SKME) and Noatikl Music Engine (NME).

The IME composes generative music based on a combination of proprietary heuristics and chance (chance based music is also sometimes referred to as being aleatoric, stochastic or algorithmic). It creates MIDI notes and MIDI controller events through the interaction of 4 different kinds of Objects: Generator (with 7 Generator Types), Rule (Scale, Harmony, Rhythm & Next Note), Cell and File. There can be multiple instances of the first two objects in a "Cell" and there are a large number of Generator parameters to affect and direct the composition.

It is referred to as being 'adaptive' because its "Listening Generator" allows external MIDI notes to influence a live composition.

Apps that include the IME 21:

- Wotja 21 - See the Wotja 21 User Guide

Sound Generation

The IME MIDI data can (as app available or permitted) be used to drive external MIDI software or hardware synths, FX units and samplers or can be played through the Intermorphic Sound Engine (ISE) for truly portable generative music.

Next Steps

Fundamental to understanding how the IME works is understanding two types of Objects: Generator and Rule. We mention these right at the start of this guide, as once you get your head around them, you are on your way!

IME 21 Objects & Parameters

^

Generator Objects

Overview

Rule Objects

Overview

Generators

- Rhythmic
- Ambient
- Follower
- Repeater
- Patterns
- Text to Music
- Listener

Other Parameters

- Chords
- Rules
- Scripting
- Comments
- Articulation
- Controllers
- Micro Controllers
- Micro Note Delay
- Micro Pitch
- Note to MIDI CC
- Envelopes

Note: For a better (visual) understanding of how all the Generator Objects in a Cell are associated with MIDI channels and how these Generators are linked to each other (e.g. Following Generators [below], where one Generator follows the output of another) then see the Wotja Generator Network Editor.

Generator Objects

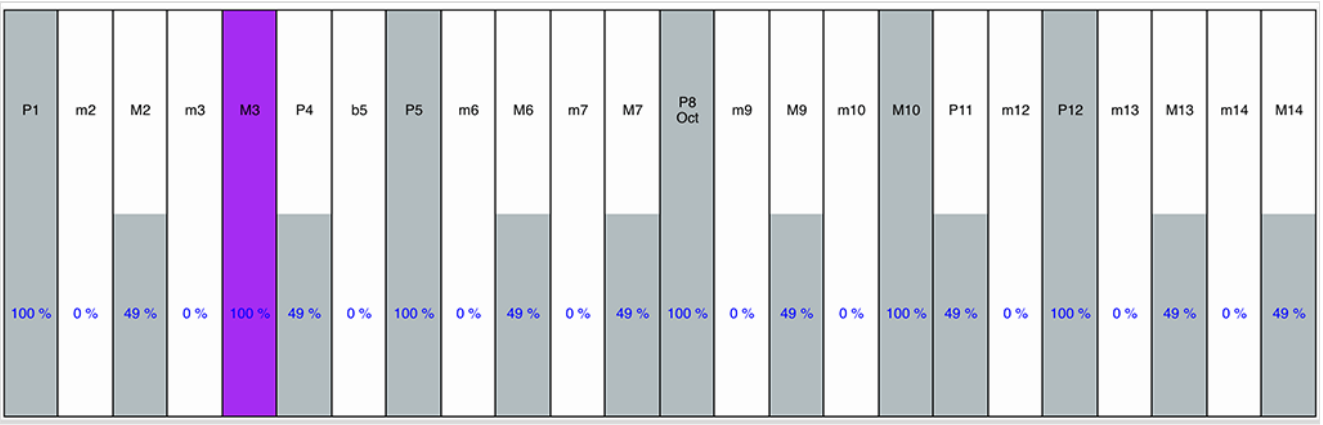
See also the Generator Editor.

Every Wotja Mix Cell includes one or more "Generator" Objects. There are 7 "types" and each generates MIDI notes and events according to various parameter values:

- Rhythmic: Note durations set by Rhythm Rules
- Ambient: Note durations set by the Ambient parameters
- Follower: Mirrors notes generated by another Generator
- Repeater: Can repeat notes it has earlier generated
- Patterns: Uses pre-defined note patterns to generate notes
- Text to Music: Generates notes from text input
- Listener: Detects incoming MIDI notes/events

Tip: The Generator button will flash when the Generator plays.

Note: For historical reasons these used to be called "Voices".



Rule Object - Scale Rules

See also the Rule Editor and Mix Rules and Cell Rules.

Every Generator composes notes according to the settings of 4 types of Rule Objects it has associated with it: Scale Rule, Harmony Rule, Next Note Rule, Rhythm Rule. Each Rule Object contains one or more "Rules", such as "Major" or "Minor".

Each Rule is an array of the values of the elements for that Rule Object (such as the values of 24 scale intervals). These values represent **the relative probability (from 0% to 100%)** (the "weighting") of that Rules' element being chosen by a Generator for a note to be composed. An element with a value of 0 will *not* be chosen for composition.

Important: When the Pitch Range for a Generator is wide enough, Scale Rules will "wrap around" to accomodate the extra range in notes available.

Changes to the values in these Rules, as with changes to any other IME parameter, allows an instant change in the music generated.

Tip: Where the interface displays the Rule elements, tap or click one one at the appropriate position to change the value of that element; the IME will attempt to make that element flash when it has been selected to play a note.

Generator - Rhythmic ^

Overview

A "Rhythmic" Generator is the default Generator Type. It composes according to the parameters in the "Basics" Parameter Group. The group has this name because the parameters in it (e.g. MIDI Channel, Patch etc) are common to and used by all Generators.

Tip: To best allow note durations to fit within the bar structure of a Cell, a Rhythmic Generator calculates them according to the Rhythm Rule it uses.

Parameter Group - Basics

Name
Mute
MIDI Channel
Send MIDI Bank/Patch?
Patch
Pitch
Pitch Range
Phrase Length
Phrase Length Range
Phrase Gaps
Phrase Gaps Range
Note Rest %

Name ^

Every Generator in a Noatikl file has a unique name. You can use any name you want, provided it is not empty, and provided it is not a single question mark (which has a reserved meaning for use with Rules, which you will find out about later).

Mute ^

Toggle this setting (Yes/No) to mute or unmute the Generator. Certain Generator Types might take some time to respond, depending on how far in advance their notes are composed.

When the keyboard focus is on the Mute cell, you have various extra menu options available to you in the "Control" Menu. These are as follows:

- Solo Generator
- Unmute All Generators
- Mute All Generators

If you hold down the *ctrl* key when you click on the mute cell, you will toggle all other Generator's mute states, without changing the mute state of the Generator that you *ctrl-click* on. This can be very handy.

MIDI Channel ^

A Generator emits data on a MIDI Channel. MIDI channels are numbered from 1 to 16. The default MIDI channel for a Generator is actually MIDI channel 0 – which tells the IME to assign a free channel from 1 to 16 automatically, as best it can. MIDI channel 10 is always reserved for percussion sounds, such as drum sounds or other untuned sound.

Send MIDI Bank/Patch? ^

Not all software synthesizers for your favourite sequencer like having Patch data supplied to them via a Patch Change MIDI event. If this is the case, simply change the *Send MIDI Bank/Patch?* parameter to *No* (unchecked), and the IME won't send any MIDI patch change events.

Patch ^

Every Generator is assigned a given Patch. This specifies the sound that you will hear whenever the Generator plays a note. The exact sound you hear depends on the sound source you have associated with the Generator; this could be the ISE or you might be driving an external MIDI synth.

In general, the IME does not emit any MIDI bank select CC information for a Generator before it emits the Patch Change MIDI event. However, you *can* force the IME to emit such information, by typing-in a special format patch value; where you type-in the patch in the format: *patch.msb.lsb*, for example:

```
98.53.4
```

In this example, the IME will emit bank select CCs for both MSB and LSB according to the settings you supply (53 and 4 respectively, in this case). If you don't specify a value for the lsb, then the IME will only emit a Bank Select MSB CC (CC number 0). If you supply the lsb, then the IME will also emit a Bank Select MSB CC (CC number 32).

Pitch ^

Set the Pitch to be the minimum pitch for which you want your Generator to compose. The IME will ensure that it composes no notes less than this pitch value.

Pitch Range ^

Set the range of pitches in semitones above the Pitch which can be used for composition. The IME will ensure that it only composes notes with pitches between Pitch and Pitch + Pitch Range.

Phrase Length ^

Set this to define the shortest possible sequence of notes that your Generator will compose in sequence. The Generator composes a sequence of notes, followed by a sequence of rests. The length of each sequence of notes is governed by this and the Phrase Length Range parameter.

Phrase Length Range ^

This value defines the upper limit to the number of notes that your Generator will compose in sequence. For example, if the Phrase Length is 3, and the Phrase Length Range is 25, then the minimum phrase will be 3 notes, and the maximum phrase length will be (3+25) = 28 notes.

Phrase Gaps ^

Set this to define the shortest possible sequence of rests that your Generator will compose. Your Generator composes a sequence of notes, followed by a sequence of rests. The length of each sequence of rests is governed by this and the Phrase Gaps Range parameter.

Phrase Gaps Range ^

This value defines the upper limit to the number of rests that your Generator will compose in sequence. For example, if the Phrase Gaps is 3, and the Phrase Gaps Range is 25, then the minimum phrase will be 3 rests, and the maximum phrase length will be (3+25) = 30 rests.

Note Rest % ^

This value defaults to zero. If not zero, then the defined percentage of notes that would otherwise be played by your Generator will instead be treated as a rests of the same duration. This is very useful for making any Generator sound sparser. Give it a go: this parameter is very powerful, and applies to **all** Generator Types.

Generator - Ambient

Overview

Tip: See also Rhythmic for the common "Basics" Generator parameters.

An "Ambient" Generator does not use the Rhythm rule for note durations but instead use durations defined by the specialist parameters in the Ambient parameter group. This is to allow them to generate notes without respect for tempo or bar timings. They are wonderful for creating drifting, floating sounds for either background or foreground use as drones or for musical texture.

Parameter Group - Ambient

Units
Duration
Duration Range
Gap Minimum
Gap Range

Units

You define the Unit of Measure for which the other Ambient Generator parameters are interpreted. This may be one of the following values:

- *Seconds (thousandths of a)*
  The parameters including Duration are all interpreted as being in thousandths of a second (i.e. Milliseconds). So, a Duration value of 1000 means one second.
- *Beats (60ths of a)*
  The parameters including Duration are all interpreted as being in 60ths of a beat. In the IME a Beat is defined as being one crotchet; you get 4 beats in a bar of 4:4 music. So, a Duration value of 60 means one beat. A Duration value of 30 means a quaver. A Duration value of 20 means a triplet. A Duration value of 15 means a semi-quaver. A Duration value of 240 means 4 beats (which is a full bar if the Cell Meter is 4:4).
- *Full seconds*
  The parameters including Duration are all interpreted as being in seconds. So, a Duration value of 10 means ten seconds.

Duration

The Ambient Generator parameters govern how Ambient Generators work.

This defines the minimum duration for which the Ambient Generator will play when it composes a note. The actual value chosen for each note is a value between Duration, and Duration plus the Duration Range. Each and every note composed for this Ambient Generator will have a note whose duration is separately calculated.

Duration Range

This is combined with the Duration parameter, to determine the duration for which the Ambient Generator will play when it composes a note. The actual value chosen for each note is a value between Duration, and Duration plus the Duration Range. Each and every note composed for this Ambient Generator will have a note whose duration is separately calculated.

Gap Minimum ^

This defines the minimum duration for which the Ambient Generator will play when it composes a rest. The actual value chosen for each rest is a value between *Gap Minimum*, and *Gap Minimum* plus the *Gap Range*. Each and every rest composed for this Ambient Generator will have a rest whose duration is separately calculated.

Tip: This is the Duration used for the Phrase Gaps / Phrase Gaps Range parameters i.e. it allows the duration of gaps to be different to that of the notes.

Gap Range ^

This is combined with the *Gap Minimum* parameter, to determine the duration for which the Ambient Generator will play when it composes a rest. The actual value chosen for each rest is a value between *Gap Minimum*, and *Gap Minimum* plus the *Gap Range*. Each and every note composed for this Ambient Generator will have a note whose duration is separately calculated.

Tip: This is the Duration Range used for the Phrase Gaps / Phrase Gaps Range parameters i.e. it allows the duration of gaps to be different to that of the notes.

Generator - Follower ^

Overview

A "Follower" Generator is used in a call-response manner, allowing it to follow the behaviour of other Generators according to parameters in the "Following" Parameter Group.

In the Cell, select the Generator which you want your "Following" Generator to follow - you may follow any Generator of any type. You may even follow a Generator that follows another Generator that follows another Generator... provided that you don't try to define a cyclic dependency which loops back to the current Generator! If you *don't* specify a Generator to follow then it won't play.

Important: If you follow a Generator that is using the Chords parameter (i.e. to create more than one note), then only the first note in the chord is followed.

Parameter Group - Follower

Follow Generator?	Delay Generator?
Percent	Delay Range
Strategy	Shift Interval
Units	Shift Interval Range

Follow Generator? ^

This parameter simply allows you to select another Generator *in the Cell* for your "Following" Generator to follow.

Percent ^

This parameter sets the percentage of notes that the Followed Generator responds to so as to emit a note. Set to 100 if you want the Following Generator to emit a note for every note played by the Followed Generator. Set to a smaller value if you want to thin-out the notes played by the Following Generator. This is also useful for building networks of chords, where if you have a number of Following Generators all following either each other or one main Generator, and if those Following Generators have the Percent Parameter to less than 100, then sometimes you will hear dense chords, and sometimes you will hear thinner chords.

Strategy ^

This parameter defines the pitch to use for a note generated by the Following Generator. The available values are:

- *Chordal Harmony*
  This causes the Following Generator to choose notes which respect the currently defined Scale, Harmony and Next Note Rules.
- *Interval Within Scale Rule*

This causes the Following Generator to choose notes which are offset from the followed note, such they are at an interval within the Scale Rule, defined as a value randomly selected between the Shift / Interval and Shift / Interval plus Shift / Interval Range values.

For example, if these values are 1 and 2 respectively, then each time a note is chosen, it will be between 1 and (1+2)=3 Scale Rule intervals up from the Followed Generator's note. It is important to understand that this refers to the non-zeroed elements in the current Scale Rule, in other words only those notes that are available within the Scale Rule.

So, in our example, if we were using a Major Scale Rule, and if the followed note were C4 (Middle C), and if Noatikl chose a value of 2 as its random value; then the played note would be E4 (Middle C), which is the second note up from Middle C within the Major Scale Rule.

- *Semitone Shift*
  This causes the Following Generator to choose notes which are offset up from the followed note, such they offset from the followed note by a number of semitones which is a value randomly selected between the Shift / Interval and Shift / Interval plus Shift / Interval Range values. A note chosen in this way ignores the current Scale Rule.

  For example, if these values are 1 and 2 respectively, then each time a note is chosen, it will be between 1 and (1+2)=3 semitones up from the Followed Generator's note.

  So, in our example, if we were using a Major Scale Rule, and if the followed note were C4 (Middle C), and if Noatikl chose a value of 3 as its random value; then the played note would be D#4 (Middle D#), which is the third semitone note up from Middle C. This value is used even though it is not in the current scale rule.

Units ^

You define the Unit of Measure by which the Delay and Delay Range parameters are interpreted. This may be one of the following values:

- *Seconds (thousandths of a)*
  The parameters including Duration are all interpreted as being in thousandths of a second (i.e. Milliseconds). So, a Duration value of 1000 means one second.
- *Beats (60ths of a)*
  The parameters including Duration are all interpreted as being in 60ths of a beat. In Noatikl a Beat is defined as being one crotchet; you get 3 beats in a bar of 4:4 music. So, a Duration value of 60 means one beat. A Duration value of 30 means a quaver. A Duration value of 20 means a triplet. A Duration value of 15 means a semi-quaver. A Duration value of 240 means 4 beats (which is a full bar if the Cell Meter is 4:4).
- *Full seconds*
  The parameters including Duration are all interpreted as being in seconds. So, a Duration value of 10 means ten seconds.

Delay ^

This defines the minimum delay after which the Following Generator will play a followed note. The actual value chosen for each note is a value between Delay, and Delay plus the Delay Range. Each and every note composed for this Following Generator will have a note whose delay is separately calculated.

Delay Range ^

This is combined with the Delay parameter, to determine the delay after which the Following Generator will play a followed note. The actual value chosen for each note is a value between Delay, and Delay plus the Delay Range. Each and every note composed for this Following Generator will have a note whose delay is separately calculated.

Shift/Interval ^

Used when the Strategy is either Interval Within Scale Rule or Semitone Shift.

This causes the Following Generator to choose notes which are offset in some way from the followed note, according to the Strategy; where the offset is defined as a value randomly selected between the Shift/Interval and Shift/Interval plus Shift/Interval Range values.

Shift/Interval Range ^

This represents the "*Shift/Interval Range*", and is used when the Strategy is either Interval Within Scale Rule or Semitone Shift.

This causes the Following Generator to choose notes which are offset in some way from the followed note, according to the Strategy; where the offset is defined as a value randomly selected between the Shift/Interval and Shift/Interval plus Shift/Interval Range values.

Generator - Repeater ^

Overview

"Repeater" Generators are like "Rhythmic" Generators, with the added feature that they can be defined to repeat notes that they have composed in previous bars, according to rules defined in the Repeat Parameter Group. When not repeating previous bars, Repeater Generators compose as if they were of Rhythmic Generator Type.

Parameter Group - Repeater

Repeat Generator?
Percent
Bars
Bars Range
History Generator?
History Range

Repeat Generator? ^

You define the name of the Generator from which you would like, from time-to-time, to repeat past bars of music. If you simply want to repeat bars played in the past for the current Generator, simply select the magic value of '?', which is also the default value.

Percent ^

When the Generator starts composing a new bar, it takes a look at this parameter value. This defines for what percent of the time the Generator should repeat previously-composed music. Set this parameter to 100 if you always want past composed music to be repeated (where available!); set to 0 if you never want past music repeated by this Generator. When the Generator doesn't choose to repeat past data it composes a new bar of music were it to be of Rhythmic Generator Type.

Bars ^

Defines the number of bars for which the Generator should repeat a past-composed chunk of music. The actual value chosen is somewhere between Bars and Bars + Bars Range.

Bars Range ^

Defines the upper limit of the number of bars for which the Generator should repeat a past-composed chunk of music. The actual value chosen is somewhere between Bars and Bars + Bars Range.

History ^

Defines the number of bars in the past, from which the Generator will choose the past-composed music to repeat. The actual value chosen is somewhere between History and History + History Range.

History Range ^

Defines the upper limit of the number of bars in the past, from which the Generator will choose the past-composed music to repeat. The actual value chosen is somewhere between History and History + History Range.

Generator - Patterns ^

Overview

"Patterns" Generators generate notes from various fixed MIDI-like patterns. These patterns have a specific syntax, are able to follow generative sequencing rules and can adapt automatically to changes in Scale Rules. They are great for bringing some structure to your composition. Patterns Generators are capable of mutating their patterns while playing, according to parameters defined in the Patterns Parameter Group. When mutating a pattern, Patterns Generators compose as would a Rhythmic Generator.

Parameter Group - Patterns

- Pattern List
- Use Percent
- Mutation Factor
- Bars Between
- Bars Range
- Mutate Rhythm?
- Meter

Pattern List ^

The "Pattern" parameters govern how Pattern Generators work. Pattern Generators compose as if they were of Rhythmic Generator Type. Generators using TTM (Text to Music) Generators still use the Pattern Generator Type, as TTM is a subtype of that. The easiest way to use them is in the Wotja Pattern Editor.

Pattern Syntax Overview

The pattern syntax is somewhat complicated and will require a bit of effort to get to grips with. Even if you do wish to use it, we still recommend that you first use the Wotja Pattern Editor to familiarise yourself with patterns and how they work.

A pattern is a text string in a specific format, surrounded by < and > symbols. Patterns can be grouped together as list of patterns. Because of this we refer to patterns as a "Pattern" when there is only one, or as "Sub-Patterns" when there are many in the list of patterns. We know it is a bit confusing, so we'll say it again in another way: each Pattern is made-up of a number of Sub-Patterns.

There are 4 types of Pattern:

- ***Note***
  - **R - Rhythm**: Defines note durations to use, but leaves selection of the note pitches to use up to the Generator acting as a Rhythmic Generator. A negation duration represents a rest.
  - **B - Both**: i.e. "Melodic" pattern using scale rule intervals. Includes a series of *both* paired note durations and scale rule interval values. A scale rule interval value of 1 represents the 0ve scale interval used by the Generator and a value of 0 represents a rest.

    - **F – Fixed**: i.e. "Melodic" pattern using fixed MIDI pitches. Defines the root pitch (60 is Middle C) to use for the pattern and a series of both paired note durations and relative pitches. The pitch values are relative to the root pitch and are independent of scale rule. A pitch value of 0 represents the root pitch and a value of -1 means treat this note as a rest note. These patterns can be useful for drum riffs, e.g. with MIDI drums.
- ***Sequence***
    - **S – Sequence**: Sequenced patterns allow the IME to use generative rules to select which sub patterns to use while playing a pattern as a generative sequence of sub-patterns.

The IME employs an underlying "time unit" that is 1/60th of a crotchet/quarter note. IME note Duration values map onto standard music notation in the following way (irrespective of meter) [# IME time units - Composed note length]:

- 240 - whole note (i.e. one bar of 4:4)
- 120 - minim/half note
- 60 - crotchet/quarter note
- 30 - quaver/eighth note
- 20 - triplet/twelfth note
- 15 - semi-quaver/sixteenth note

So by way of example:

- one bar of 4:4 is 4 * 60 = 240 IME time units.
- one bar of 3:4 is 3 * 60 = 180 IME time units.
- one bar of 2:4 is 2 * 60 = 120 IME time units.
- one bar of 1:4 is 1 * 60 = 60 IME time units.
- one bar of 6:8 is 6 * 30 = 180 IME time units.
- one bar of 9:8 is 9 * 30 = 270 IME time units.

You are of course free to experiment using other time unit values, which will mean different things. E.g. 10 time units is a 24th note etc.

Note Sub-Pattern Syntax:

Tip: To aid understanding and visual clarity, we use the following color coding: durations, velocity values and scale intervals.

<[prob][.M] pattype {[dur][.vff[-vffr]] [scaleint]}*>

Where:

- *prob* : the relative probability that this sub-pattern is selected; relevant only where there is more than one sub-pattern! The default value is 100%.
- *M* : Flag indicating that the sub-pattern is to be "muted", i.e. not allowed to be selected. This can be useful for testing of individual sub patterns; where you might want to "solo" a sub-pattern by muting out all the others.
- *pattype* : One of:
    - R: Rhythm
    - B: Both
    - F: Fixed, followed by the root note in MIDIpitch. e.g. F60.

- *dur* : Note Duration / IME time units.
  - See above for note duration values.

    In R patterns a negative duration indicates a rest for that time; in F patterns a pitch of -1 indicates a rest for the note duration.

    If a note sub-pattern is not an exact even number of bars (e.g. 2 and and half bars at the current meter!) then the engine will pad to silence to the end of the nearest bar boundary.

  - *vff*: Velocity Force Factor (optional, any integer value)
  - *vffr*: Velocity Force Factor Range (optional, any integer value and used in conjunction with the above).
  - Note: Velocity Force Factor is an **INTEGER % SCALE FACTOR** applied to the velocity, where the velocity is first determined from the velocity envelopes. For example, in a pattern [dur][.vff[-vffr]] 60.50 means a note of length 60 Noatikl time units with 50% VFF scaling, 30.15 means a note of length 30 Noatikl time units with 15% VFF scaling, and 120.100-20 means a note of length 120 Noatikl time units and with 100% VFF scaling with a range of 20%.
    - The scaled pattern note velocity can never be any less than 1.
    - The scaled pattern note velocity can never be any greater than 127.
    - Note that a pattern note velocity scaling factor of 0 always returns minimum MIDI velocity, which is 1. You cannot use a factor of 0 to "completely mute" a note.
    - Example: A Generator's velocity (min) envelope has a value of 100, with a velocity range envelope vallue of 20 - the velocity for a given note is selected randomly in that range, to be (say) 104. Where defined, the pattern note's Velocity Force Factor is used to scale that velocity, to a final value from 1 to 127. e.g. if the Velocity Force Factor is 50 (%), in this example we'd chose a final velocity value of: 104 * 50 (%) = 52. The Velocity Force Factor Range value, if defined, applies a range to the scaling factor.
- *scaleint*: Scale interval (not present for Rhythm patterns).
  - B rule: interpreted as being the first valid note in current Generator's Scale Rule; i.e. the first element in the Generator's Current Scale Rule which does not have a zero value. "1" is therefore usually the root note (c.f. the Pitch parameter). "0" has the special meaning of indicating a "REST" for the note duration.
  - F rule: distance in semitones up from the root note (so "0" means the root note). E.g. if F60 (Middle C), then a pitch value of 5 means MIDI pitch 65. A pitch value of -1 is a rest.

Sequence Sub-Pattern Syntax

<S100 R 1.20 2.1 1-2.1-4 2.1>

Syntax

<[S][prob][.M] R {[seqnum[-seqnumrange].[repeattimes[-repeattimesrange]] [seqnum[-seqnumrange].[repeattimes[-repeattimesrange]]}* >

Where:

- *S* : identifies a sequence sub-pattern
- *prob* : relative probability of being chosen when there is more than one sequence sub-pattern.
- *M*: Mute the sub-pattern (i.e. prevent it being selected!). If none can be selected, then a non-sequence sub-pattern is chosen to play at random as usual each time.

- *R*: Rhythm pattern type (always required)
- *seqnum*: Sequence Number.
  - The index of the non-sequence sub-pattern to play. Default is 1. The sub-patterns are numbered from 1 up.
  - *seqnumrange* : Sequence Number Range. Default is 0.
- *repeattimes*: Repeat Times Minimum.
  - The number of times to repeat this sub-pattern, when selected. A value of "0", will cause the sub-pattern (when selected) to keep playing forever until the end of the Cell! Default is 1.
  - repeattimesrange : Repeat Times Range. Default is 0.

Which sub-pattern is chosen by the Generator depends on a few things:

If there is at least one sequenced sub pattern, then a sequence is used to drive the sub-pattern. Which sequence to use, is based on the Generator making a weighted random selection from the available sub-patterns. When (if!) the sequenced sub-pattern end is reached, the Generator will make another selection as to which sequenced sub-pattern to use.

Otherwise, a sub-pattern is chosen, based on the Generator making a weighted random selection from the available sub-patterns. This sub-pattern is played through to the end, at which point the Generator will make another selection as to which sub-pattern to play.

A Note Sub-Pattern that is less than a whole number of bars at the Generator's current Meter, will be padded automatically with silence to ensure that it remains bar synchronised.

Patterns: Examples

The IME has its own pattern format, which allows it to play specified notes and rests in different ways. Patterns are affected by the various Rules being used by the Generator.

Tip: To aid understanding and visual clarity in the examples, we use the following color coding: durations, velocity values and scale intervals.

Below are a number of example patterns. Copy and paste these into Noatikl to try them out.

Rhythm: <100 R 60 60 60.127 60>

Both: <100 B 60.15-30 1 60 2 60.127 3 15 7>

Forced Frequency: <100 F60 60.127 1 60 4 30 5 15.70-120 7>

Sequence: <S100.M R 1.1 2-0.1-0 3-0.1-0 >

Two patterns. Select randomly from these two each time!

<100 B 60 1 60 2 60 3 60 4>

<100 B 30 9 30 8 30 7 30 6 30 5 30 4 30 3 30 2>

Forced Frequency pattern. Plays note with pitch 58 (or MIDI patch 59 on Ch10): note on, note off, note on, note off etc.

<100 F58 15 1 45 -1 15 1 45 -1 15 1 45 -1>

Forced Frequency pattern: Plays a series of notes for a whole bar (or patches in the drum kit on Ch10) starting at base pitch 40, each one a 16th note...

<100 F40 15 1 15 7 15 6 15 13 15 2 15 8 15 4 15 11 15 13 15 2 15 8 15 2 15 2 15 8 15 4 15 14>

Two sequenced sub-patterns. Play 1 once, then 2 once...

<S100 R 1.1 2.1>

<100 B 60 1 60 2 60 3 60 4>

<100 B 30 9 30 8 30 7 30 6 30 5 30 4 30 3 30 2>

Two sequenced sub-patterns. Play 1 twice, then 2 twice...

<S100 R 1.2 2.2>

<100 B 60 1 60 2 60 3 60 4>

<100 B 30 9 30 8 30 7 30 6 30 5 30 4 30 3 30 2>

Two sequenced sub-patterns. Play 1 or 2 twice, then 1 or 2 twice...

<S100 R 1-1.2 1-1.2>

<100 B 60 1 60 2 60 3 60 4>

<100 B 30 9 30 8 30 7 30 6 30 5 30 4 30 3 30 2>

Two sequenced sub-patterns. Play 1 once, then 2 twice, the one or 2 once, then 2 once...

<S100 R 1.1 2.2 1-1.1 2.1>

<100 B 60 1 60 2 60 3 60 4>

<100 B 30 9 30 8 30 7 30 6 30 5 30 4 30 3 30 2>

Two sequenced sub-patterns. Play 1 once, then 2 forever...

<S100 R 1.1 2.0>

<100 B 60 1 60 2 60 3 60 4>

<100 B 30 9 30 8 30 7 30 6 30 5 30 4 30 3 30 2>

Use Percent ^

When the Generator starts a new sub-pattern at the start of a bar, it consults the value you have defined for *Use Percent*. This parameter determines the probably of the Generator using the Pattern for the bar; or alternatively, compose a *completely* new bar (that you will hear only once!) were it to be of Rhythmic Generator Type.

If *Use Percent* is 100, then the Generator will always use the pattern. If *Use Percent* is 50, then teh Generator will instead compose a new bar every other bar or so. Note that the Generator will never interrupt a sub-pattern that it is playing; the *Use Percent* parameter is considered only on a sub-pattern boundary, at the start of a new bar.

Mutation Factor ^

The mutation factor is used when a bar is considered for mutation (which can happen only if *Bars Between* is not zero! The *Mutation Factor* determines the level of mutation to apply. If set to 10.0%, then when playing from a sub-pattern, this means that each note that would be played from the pattern, has a 10% chance of having a different one composed, with subsequence pattern playbacks keeping that mutation. Note that if *Mutate Rhythm?* is set to *Yes*, then if the composed note is longer than the composed-over pattern note, this might overlap and cancel-out some other notes in the sub-pattern.

Bars Between ^

This parameter defines the number of bars that the Generator waits between, before trying to mutate a bar in a pattern according to the *Mutation Factor*. If *Bars Between* is set to zero, the Generator can never mutate. Set to 1 if you want mutation every bar, 2 if you want mutation every other bar, etc. ...

The actual number of bars used is selected randomly each time, somewhere in the range from *Bars Between*, to *Bars Between* plus *Bars Range*.

Bars Range ^

This parameter is used to help define the number of bars between attempts by the Generator to mutate the current pattern. The actual number of bars used is selected randomly each time, somewhere in the range from *Bars Between*, to *Bars Between* plus *Bars Range*.

Mutate Rhythm? ^

If set to *No*, then the timing of the sub-pattern is preserved perfectly; only the frequency of the pattern notes will be changed when the pattern is mutated. Otherwise, the duration of each note is chosen from the rhythm rules and phrase/phrase gap rules for the Generator.

Meter ^

Defines the Meter to be used by the Generator, such 4:4 or 3:4 or 6:8. A value of ?, which is the default, means to use the Meter defined for the Cell. A different value allows the Generator to work with a completely different meter, which can be used for interesting polyphonic effects.

Generator - Text to Music (TTM) ^

Overview

"Text to Music" Generators are new Generator Type in V20. Text to Music Generators have their own special set of parameters and allow text *in any language* to generate a seed melody which, even though not visible, is in a Pattern syntax. Use English, Chinese, Japanese, Russian, German, French - whatever language takes your fancy, you will always get a melody!

Parameter Group - Text to Music (TTM)

Cut-up Rule
TTM Text
Display?
Repeats
Repeats Range
Tune Start at Index
Tune Length Override
Phrase Length
Phrase Length Range
Gaps
Gaps Range
Interval
Interval Range
Improvise after Tune
Variaton

Cut-Up Rule ^

The selection you make here determines what text is used for TTM. If you select one of the Cut-up options then **text from your mix Cut-up** will be displayed in the TTM Text field below and will overwrite any existing Custom text. Note that if there is no text available in a selected line of Cut-up then the TTM Generator will have no text to work with.

- *Custom*: Uses text that you enter in the TTM Text field.
  - See the TTM Editor for details on how to do this.
- *Cut-Up: Line#=Track#* (Default): This setting is very powerful when it comes to mixes that feature TTM. It means you can harness the power of the Cut-up Editor to quickly generate text that can be used to populate the TTM for multiple Generators - all at once! It is easiest to understand by example: if you have this setting and are using a TTM Generator in say Track 1 then it will use line 1 of text in your mix Cut-Up. If you had a TTM Generator in Track 3 it would use line 3 of your mix cut-up, and so on.
- *Cut-up: All Lines*: Uses the text in all lines in your Cut-up for the purposes of TTM
- *Cut-Up: Line# 1-11*: Uses the text from Line# 1 to Line# 11 (as selected) of your Cut-up for the purposes of TTM. You can set up TTM generators all to use the same line if you want!

TTM Text ^

This is the text *in any language* used to generate your TTM seed melody. In general it takes 2 characters to generate a note. Note that if your text is < 6 characters (e.g. Hello, which is 5 characters), we turn it into HelloHello ... which makes it sufficiently long to make at least 3 notes (that actually generates a few more). In Wotja, tap on this field to go to the TTM Text Editor screen.

Display? ^

The Display? toggle determines if your TTM text will display in Fullscreen mode.

If this is toggled ON, and IF in Display Mode you have selected Text > Show Text from TTM Generators then the TTM Text above will display on screen when in Full Screen or in Display Mode. If it is OFF, then it will not display.

Repeats ^

The total number of times the original melody or a variation of it is played. This also applies to improvised melodies (see Improvise toggle).

Repeat Range ^

Sets the range above the minimum. Also applies to improvised melodies (see Improvise toggle).

Tune Start at Index ^

From the notes composed, set the first note you want your tune to start playing at. Maximum value is Notes - 1.

Tune Length Override ^

From the notes composed, set how many notes will play. Maximum value is Notes - Tune Start at Index.

Phrase Length ^

Defines the minimum number of notes there are in a "phrase" (and you can see in the indicator above how many notes your text has generated - all of these notes get put into phrases to make the "tune").

Phrase Length Range ^

Sets the range above the minimum.

Gaps ^

Defines the minimum number of rests between each phrase (allows a tune to breathe). Rests are measured in terms of 16th notes.

Gaps Range ^

Sets the range above the minimum.

Interval ^

Defines the minimum number of rests between each play of the tune (allows a tune to breathe). Rests are measured in terms of 16th notes.

Interval Range

Sets the range above the minimum.

Improvise after Tune

Turn this on to let Wotja continuously generate tune variations after the FIRST complete play (and repeats) of the tune. When this is on, Wotja can "noodle" for ever (well, until you turn it off or the Sleep Timer kicks in, if set!).

Tip: If you have this set to ON and Variation (below) set to 0, then the TTM melody will continue to repeat for ever.

Variation

Selects how much variation is applied to the previous melody when improvising.

Overview

A "Listener" Generator detects a monophonic (not polyphonic) incoming MIDI note event on its MIDI channel and presents it as a virtual composed note. It will only detect a subsequent note if it first receives a note off for the previous note.

To generate sound via the ISE it needs to be followed by a Follower Generator. Used like this, Listener Generators allow you to create simple hyperinstruments where external input can influence the music that is generated. Such music is also referred to as **Adaptive Generative Music [AGM]**.

Listener Generators do not have any special parameters, which is why there is no special Parameter Group, and nor do they require any use of scripting. However, they can be used with Intermorphic Wotja Script to create more advanced hyperinstruments.

When an incoming MIDI note is detected, e.g. C60, then:

1. If the Generator has a Chords Depth value of 1 and Chords Range value of 0 (both default values) it echoes a single virtual note that can be "heard" only by a Follower Generator - this is used to create an actual note.
2. If the combined Chords Depth or calculated Chords Depth/Chords Depth Range value is >1 then the first note is still silent (and echoed as above and so can be followed, too), but the composed chording notes will also play (e.g. through the ISE) and Generators can follow those notes, too.

The virtual note created by a Listener Generator is pitch shifted, if necessary, to fit within the band of pitch values set by its Pitch and Pitch Range parameters (see Rhythmic Generator).

Playing with a Listener Generator:

- Ensure Wotja has been setup, as below.
- Ensure your Listener Generator is on the MIDI channel you are expecting to detect MIDI notes.
- Either Follow that Listener Generator with a Follower Generator (1 above) and maybe use that Generator's Chording parameters to play chords, OR set its Chording parameters to >1 (as in 2 above).
- Press play in Wotja app to start your Cell and for the Listener Generator to detect the MIDI notes fed into it (and remember that the Listener Generator will only detect a subsequent note if it first receives a note off for the previous note!).

Wotja Setup:

MIDI emitting app or MIDI Device feeding MIDI notes into the Wotja app

- Load the MIDI app (e.g. for iOS MIDIKeys or e.g. for macOS MIDI Mock or Live Performer) and ensure you have it set up to send MIDI notes on the Virtual MIDI channel you want, e.g. MIDI Channel 1 or Omni etc. Tip: In MIDIKeys you must have CoreMIDI Port set to 'Virtual Port'.
- In Wotja > Settings, select the MIDI Input Devices button which takes you to the MIDI Input Devices screen:
  - Toggle on the relevant Channels for your desired MIDI Input Device(s), e.g. 'Network Session 1' on iOS, or e.g. 'MIDIKeys'.
- To hear sound in Wotja, ensure you have the "ISE for Sounds & FX" setting toggled on in Settings. Note that when playing through the ISE the latency you will experience is mostly affected by both the Audio Block

Size. When that is 1024 samples the latency is about 1 second; when set to its minimum value of 256 samples then it is about 1/4 of a second. You may also wish to experiment with the MIDI Latency setting.

Generator - Chords ^

Overview

The Chord parameters let you configure *any* Generator Type to generate chords automatically.

In outline, use the *Depth* and *Depth Range* values to define the "Chord Depth"; which is the number of notes that will play at any one time for a given Generator. The first note in any chord is composed according to the normal mechanism for the Generator Type; additional notes that cause a chord to be built-up may be generated automatically according to the Chord parameters.

Parameter Group - Chords

Depth
Depth Range
Percent
Strategy
Units
Delay
Delay Range
Shift/Interval
Shift/Interval Range
Pitch Offset
Velocity Factor

Depth ^

Specify the minimum *Depth* of chord that you want your Generator to play with. A value of 1 will mean that the Generator will not chord (unless the *Depth Range* parameter is greater than zero).

The *Depth* defines the number of notes that are played by the Generator at any one time.

Depth Range ^

Specify the relative maximum *Depth* of chord that you want your Generator to play with. A value of 0 means that whenever the Generator is played, it will play a number of notes equal to the *Depth*. A value of one or more means that whenever the Generator is played, it will play a number of notes equal to a randomly selected value between the *Depth* and the *Depth* plus the *Depth Range*.

Percent ^

This parameter tells the Generator the percentage chance that it should actually emit any given note in the chord (after the first note, of course!). Set to 100 if you want the Chording Generator to always emit a note for every note played by the Generator. Set to a smaller value if you want to thin-out the notes played within the chord. This allows you to create chords of varying depth; sometimes dense, sometimes thin.

Strategy ^

This parameter tells the Generator what it should do when it decides the pitch to use for a note generated within a chord. The available values are:

- *Chordal Harmony*
  This causes the Generator's chord notes to be selected according to the currently defined Scale, Harmony and Next Note Rules.

- *Interval Within Scale Rule*
  This causes the Generator's chord note to be selected offset from the followed note, such they it is at an interval within the Scale Rule beyond the previous note in the chord, defined as a value randomly selected between the Shift / Interval and Shift / Interval plus Shift / Interval Range values.

  For example, if these values are 1 and 2 respectively, then each time a note is chosen within the chord, it will be between 1 and (1+2)=3 Scale Rule intervals up from the previous note in the chord. It is important to understand that this refers to the non-zeroed elements in the current Scale Rule, in other words only those notes that are available within the Scale Rule.

  So, in our example, if we were using a Major Scale Rule, and if the first note in the chord were C4 (Middle C), and if the Generator chose a value of 2 as its random value; then the played note would be E4 (Middle C), which is the second note up from Middle C within the Major Scale Rule.

- *Semitone Shift*
  This causes the the Generator's chord note to be selected offset up from the previous note in the chord, such it is offset from the previous chord note by a number of semitones which is a value randomly selected between the Shift / Interval and Shift / Interval plus Shift / Interval Range values. A note chosen in this way ignores the current Scale Rule.

  For example, if these values are 1 and 2 respectively, then each time a note is chosen, it will be between 1 and (1+2)=3 semitones up from the previous note in the chord.

  So, in our example, if we were using a Major Scale Rule, and if the previous note in the chord were C4 (Middle C), and if the Generator chose a value of 3 as its random value; then the played note would be D#4 (Middle D#), which is the third semitone note up from Middle C. This value is used even though it is not in the current Scale Rule.

Units ^

You define the Unit of Measure by which the Delay and Delay Range parameters are interpreted. This may be one of the following values:

- *Seconds (thousandths of a)*
  The parameters including Duration are all interpreted as being in thousandths of a second (i.e. Milliseconds). So, a Duration value of 1000 means one second.

- *Beats (60ths of a)*
  The parameters including Duration are all interpreted as being in 60ths of a beat. In the IME a Beat is defined as being one crotchet; you get 3 beats in a bar of 4:4 music. So, a Duration value of 60 means one beat. A Duration value of 30 means a quaver. A Duration value of 20 means a triplet. A Duration value of 15 means a semi-quaver. A Duration value of 240 means 4 beats (which is a full bar if the Cell Meter is 4:4).

- *Quantized Beats (60ths of a)*

This works the same way as *Beats (60ths of a)* except that where the *Delay* has a special value of 10, 15 or 20; the delay is interpreted in a special way that is very useful for some breakbeat-based music. Specifically, in this case, the calculated value for the delay is rounded to the nearest sub-multiple of the *Delay* value. So, for example, if the engine calculates a value of 43, and if *Delay* is 20, the used value for the delay is actually 40 (which is the nearest multiple of 20).

Delay ^

This defines the minimum delay after which the Chording Generator will play a followed note. The actual value chosen for each note is a value between Delay, and Delay plus the Delay Range. Each and every note composed for this Chording Generator will have a note whose delay is separately calculated.

Delay Range ^

This is combined with the Delay parameter, to determine the delay after which the Chording Generator will play a followed note. The actual value chosen for each note is a value between Delay, and Delay plus the Delay Range. Each and every note composed for this Chording Generator will have a note whose delay is separately calculated.

Shift/Interval ^

Used when the Strategy is either Interval Within Scale Rule or Semitone Shift.

This causes the Chording Generator to choose notes which are offset in some way from the followed note, according to the Strategy; where the offset is defined as a value randomly selected between the Shift / Interval and Shift / Interval plus Shift / Interval Range values.

Shift/Interval Range ^

This represents the "*Shift/Interval Range*", and is used when the Strategy is either Interval Within Scale Rule or Semitone Shift.

This causes the Chording Generator to choose notes which are offset in some way from the followed note, according to the Strategy; where the offset is defined as a value randomly selected between the Shift / Interval and Shift / Interval plus Shift / Interval Range values.

Pitch Offset ^

This parameter defines the amount that the pitch of each note in the chord should be offset, in semitones, from the previous note in the chord; the actual value selected might be overridden according to the various rules that apply to the Generator, but in general, this parameter allows you to "shape" a chord to have a given range of pitch values. In combination with the *Delay*-related parameters, this allows you to create some very interesting arpeggiation effects.

For example, a value of +12 would tend to space each note in the chord by a range of 12 semitones (which is one octave), with each subsequent value in the chord being higher in pitch that the previous.

For example, a value of -12 would tend to space each note in the chord by a range of 12 semitones (which is one octave), with each subsequent value in the chord being lower in pitch that the previous.

Velocity Factor ^

This parameter allows you to specify the range of relative velocities for the notes in a chord. Each subsequent note in the chord is the defined percentage louder (for a positive value) or quieter (for a negative value) than the previous note in the chord. A value of zero means that all notes in the chord are played with the same velocity.

The Generator velocity envelope values are ignored when this parameter is applied.

For example, a value of *-30* would tell the Generator to generate its chords such that each auto-chorded note is 30% quieter than each preceding note in the chord; giving a noticeable tailing-off effect.

Generator - Rules ^

See also the Rule Editor.

The Generator Rule parameters let you select the various rules that govern how your Generator works. The Rules themselves are edited in the relevant Rule Object. See Rule Objects.

Parameter Group - Rules

Scale
Harmony
Rhythm
Next Note
Harmonize?
Generator Root

Scale Rule ^

See also: Rule Editor and See Rule Objects; Standard Rule Values

Each Generator composes according a Scale Rule. The Rule defines both the notes in the musical scale that are available for use and the note probability weightings.

Scale Rules can selected from the Standard Rule Values list (below) or can be a custom Rule you have created and named as you want.

Scale Rules can be set at a Generator, Cell or Mix level - see the Rule Editor for details.

- **Scale Rule elements: 24**
  - These elements represent the semitone distance from the Root note defined for this Cell, these being: P1 (root), m2, M2, m3, M3, P4 (perfect 4th), b5, P5 (perfect 5th), m6, M6, m7, M7, P8 Oct (octave), m9, M9, m10, M10, P11 (perfect 11th), m125, P15 (perfect 13th), m14, M14, m15, M15.
  - Element values: % probability weighting
- If opening an "old" rule that contains just one octave, we automatically extend it by duplicating into a higher octave, and;
- We wrap around the two octaves, matching Pitch Minimum as well as we can:
  - If Generator has pitch minimum of B, but Root is C; wrap around the rule starting from the C below.
  - If Generator has pitch minimum of D, but Root is C; wrap around the rule starting from the D below.

Included Scale Rules

The following "Standard Rule" items are included (Note: Rule objects can be referenced in Wotja Script):

Scale Rule Name	Element Values
Default	1 0 0.5 0 1 0.5 0 1 0 0.5 0 0.5
All Scale Major	1 0 1 0 1 1 0 1 0 1 0 1

All Scale Minor Natural	1 0 1 1 0 1 0 1 1 0 1 0
All Scale Minor Harmonic	1 0 1 1 0 1 0 1 0 0 0 1
Chord Major triad	1 0 0 0 1 0 0 1 0 0 0 0
Chord Major sixth	1 0 0 0 1 0 0 1 0 1 0 0
Chord Dominant seventh	1 0 0 0 1 0 0 1 0 0 1 0
Chord Major seventh	1 0 0 0 1 0 0 1 0 0 0 1
Chord Augmented triad	1 0 0 0 1 0 0 0 1 0 0 0
Chord Augmented seventh	1 0 0 0 1 0 0 0 1 0 1 0
Chord Minor triad	1 0 0 1 0 0 0 1 0 0 0 0
Chord Minor sixth	1 0 0 1 0 0 0 1 0 1 0 0
Chord Minor seventh	1 0 0 1 0 0 0 1 0 0 1 0
Chord Minor-major seventh	1 0 0 1 0 0 0 1 0 0 0 1
Chord Diminished triad	1 0 0 1 0 0 1 0 0 0 0 0
Chord Diminished seventh	1 0 0 1 0 0 1 0 0 1 0 0
Chord Half-diminished seventh	1 0 0 1 0 0 1 0 0 0 1 0
Chord Augmented major seventh	1 0 0 0 1 0 0 0 1 0 0 1
Chord Seven-six	1 0 0 0 1 0 0 1 0 0 1 0
Chord Mixed-third	1 0 0 1 1 0 0 1 0 0 0 0
Chord Suspended fourth	1 0 0 0 0 1 0 1 0 0 0 0
Chord Dominant ninth	1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0
Chord Dominant eleventh	1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0
Chord Dominant thirteenth	1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0
Chord Seventh minor ninth	1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
Chord Seventh sharp ninth	1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0
Chord Seventh augmented eleventh	1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0
Chord Seventh diminished thirteenth	1 0 0 0 1 0 0 1 0 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0
Chord Add nine	1 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
Chord Add fourth	1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
Chord Add sixth	1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Chord Six-nine	1 0 0 0 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0

Chord Suspended second	1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Chord Jazz sus	1 0 0 0 0 1 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0
Mode Ionian (I)	1 0 1 0 1 1 0 1 0 1 0 1
Mode Dorian (II)	1 0 1 1 0 1 0 1 0 1 1 0
Mode Phrygian (III)	1 1 0 1 0 1 0 1 1 0 1 0
Mode Lydian (IV)	1 0 1 0 1 0 1 1 0 1 0 1
Mode Mixolydian (V)	1 0 1 0 1 1 0 1 0 1 1 0
Mode Aeolian (VI)	1 0 1 1 0 1 0 1 1 0 1 0
Mode Locrian (VII)	1 1 0 1 0 1 1 0 1 0 1 0
Special Root only	1 0 0 0 0 0 0 0 0 0 0 0
Special Scale Chromatic	1 1 1 1 1 1 1 1 1 1 1 1

Harmony Rule ^

See also: Rule Editor and See Rule Objects; Standard Rule Values.

Each Generator composes according a Harmony Rule. The Rule defines how likely it is that a note for that Generator will harmonize at a given musical interval in semitones with any note already composed (and in use by) by any *other* Generator.

Harmony Rules can selected from the Standard Rule Values list (below) or can be a custom Rule you have created and named as you want.

Harmony Rules can be set at a Generator, Cell or Mix level - see the Rule Editor for details.

- **Harmony Rule elements: 24**
  - These elements represent in semitones the possible harmonies available, these being: P1 (root), m2, M2, m3, M3, P4 (perfect 4th), b5, P5 (perfect 5th), m6, M6, m7, M7, P8 Oct (octave), m9, M9, m10, M10, P11 (perfect 11th), m125, P15 (perfect 13th), m14, M14, m15, M15.
  - Element values: % probability weighting
- Note: If opening a mix that uses an "old" 12 element rule we automatically extend it by duplicating into a higher octave.
- If harmonic distance is greater than 2 octaves, wrap around the rule.
- **Example**:
  - Imagine that you have three Generators, called X, Y and Z.
  - Imagine that at some time in the Cell, the IME has already chosen to play note C for Generator X, and note G for Generator Y. At that time, the IME thinks about composing a note for Generator Z. It looks at the notes available for it to compose, and adjusts the probabilities of choosing each of those notes, by applying the Harmony Rule element values for each of those two composed notes which are active at time (i.e. notes C and G).
  - Harmonies are always calculated based on a rising direction up from the Cell Root. So, if the Cell Root is B and the IME is considering if it can compose note D for Generator Z, and it looks at the

note it needs to harmonize with which is note C for Generator X; then Noatikl figures-out the Harmony Rule values for Generator Z from the C of Generator X, up and through the Octave (i.e. E, F, F#, G, G# etc.).

Included Harmony Rules

The following "Standard Rule" items are included (Note: Rule objects can be referenced in Wotja Script):

Harmony Rule Name	Element Values
Default	.35 0 0 .65 .85 1 .65 .65 .65 .1 .1 0 .35
All (one octave)	1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
All (two octaves)	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
P1	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
P1 and P8	1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

Rhythm Rule ^

See also: Rule Editor and See Rule Objects; Standard Rule Values.

Each Generator composes note durations according a Rhythm Rule. The Rule defines how likely it is that a note for that Generator has a certain duration. This rule is combined with other factors, including the remaining length of time a Generator has left in the current bar (the IME tries to avoid having notes from non-Ambient Generators drifting across bar boundaries).

Rhythm Rules can selected from the Standard Rule Values list (below) or can be a custom Rule you have created and named as you want.

Rhythm Rules can be set at a Generator, Cell or Mix level - see the Rule Editor for details.

- **Rhythm Rule elements: 12**
  - These elements represent the permitted note durations, these being: 1, 1/2., 1/2, 1/4., 1/4, 1/8., 1/8, Triplet, 1/16
  - Element values: % probability weighting

Included Rhythm Rules

The following "Standard Rule" items are included (Note: Rule objects can be referenced in Wotja Script):

Rhythm Rule Name	Element Values
Default	.1 .25 1 .5 .5 .25 .25 0 0
All But Dotted	1 1 1 0 1 0 1 1 1
Slow	1 1 1 0 1 0 0 0 0
Fast Syncopated	0 0 0 0 0 1 1 1 1
Fast Plain	0 0 0 0 0 0 1 0 1
Very Slow	1 0 .5 0 .25 0 0 0 0

Semiquavers Only	0 0 0 0 0 0 0 0 1
MIddle	0 1 0 1 1 1 0 1 0
All	1 1 1 1 1 1 1 1 1

Next Note Rule ^

See also: Rule Editor and See Rule Objects; Standard Rule Values.

For each Generator's composed notes the Next Note Rule defines the distances in semitones available to be used between each note.

Next Note Rules can selected from the Standard Rule Values list (below) or can be a custom Rule you have created and named as you want.

Next Note Rules can be set at a Generator, Cell or Mix level - see the Rule Editor for details.

- **Next Note Rule elements: 24**
  - These elements represent in semitones the distance a new note will be from the last composed note, these being: P1 (root), m2, M2, m3, M3, P4 (perfect 4th), b5, P5 (perfect 5th), m6, M6, m7, M7, P8 Oct (octave), m9, M9, m10, M10, P11 (perfect 11th), m125, P15 (perfect 13th), m14, M14, m15, M15
  - Element values: % probability weighting
- Note: If opening a mix that uses an "old" 12 element rule, we automatically extend it by setting the "upper range" items to zero.

Included Next Note Rules

The following "Standard Rule" items are included (Note: Rule objects can be referenced in Wotja Script):

Next Note Rule Name	Element Values
Default	0.25 1 1 .7 .3 .2 .1 .1 .1 .1 .05 .02 .05 0 0 0 0 0 0 0 0 0 0 0
All (one octave)	1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
All (two octaves)	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Harmonize? ^

The default value for this parameter is "Yes", which means that the Generator will be considered for harmonisation with other Generators. Set to "No" if for some reason you do not want other Generators to harmonize with this Generator.

Generator Root ^

Normally, you want your Generator to use the Cell Root. This is represented by the value ? However, sometimes you really want to force your Generator to use a different Root note; in which case, set the Generator Root to be whatever value suits.

This allows you to work-around the following sort of problem:

Imagine that you have a sampler, wher you load-up a variety of loops against MIDI note C3 up to D3. To have your Cell drive this from a Rhythmic Generator such that the sounds you hear are not affected by changes to the Cell Root, you should set the Generator Root to e.g. C3 and your Generator will then be unaffected by changes to the Cell Root. Note that in this specific example, it would probably be a good idea to set the Harmonize? Flag to No.

Generator - Scripting

The Generator Scripting parameter allows your Generator to use Intermorphic Wotja Script. For full details please refer to the Intermorphic Wotja Script Guide.

Generator - Comments

Overview

This Generator Comments parameters allow you to store comments in your Generator, in the form of copyright information and any notes you might want to make for future reference.

Parameter Group - Comment

Copyright	Notes

Copyright

Enter the Copyright information you might want to record for the Generator. In the case of a Generator from a template pack, this might contain a copyright notice associated with that template.

Notes

Enter any detailed notes you might want to make about this Generator for future reference.

Generator - Articulation ^

Overview

Important: A Generator's Articulation parameters apply to any/all notes it composes, be that via e.g. generative parameters, patterns or text-to-music.

The Generator Articulation parameters define the percentage of the duration of composed note, i.e. they determine how long a composed note actually plays for. A long time ago the IME used to compose notes to be played "Legato" (no gap between one note and the next) - this parameter allows you to play them with a shorter duration, e.g. staccato.

Parameter Group - Articulation

Minimum	Change
Range	Change Range

Minimum ^

1 is very staccato and 100 is legato (the new default).

Range ^

Max articulation is the value of Articulation (min) + Articulation range, and is used in combination with the variation values (below).

Change ^

The minimum variation in staccato between notes, c.f. other parameters that adopt the min + range approach.

Change Range ^

The range in variation of stacatto between the notes (in addition to the min).

Generator - Controllers

Overview

The Generator Controllers parameters define some of the key MIDI controller values that are emitted by the Generator.

Parameter Group - Controllers

Damper/Hold
Harmonic Content (71)
Reverb (91)
Chorus (93)
Damper Release
Portamento (65)
MIDI Channel Sharing

Damper/Hold (64)

Set this value to other than the default of "-1", if you want to emit a Damper/Hold MIDI controller (MIDI CC 64) at the specified value on this Generator's MIDI line. This is a funny MIDI controller, with only two states; in that a value of 64 or greater activates Damper/Hold, and any value of 63 or less means to turn it off! Leave this value at the default of "-1" if you don't want the Generator to emit any information for this MIDI controller.

Harmonic Content (71)

Set this value to other than the default of "-1", if you want to emit a Harmonic Content MIDI controller (MIDI CC 71) at the specified value on this Generator's MIDI line. Leave this value at the default of "-1" if you don't want the Generator to emit any information for this MIDI controller.

Reverb (91)

Set this value to other than the default of "-1", if you want to emit a Reverb MIDI controller (MIDI CC 91) at the specified value on this Generator's MIDI line. Leave this value at the default of "-1" if you don't want the Generator to emit any information for this MIDI controller.

Chorus (93)

Set this value to other than the default of "-1", if you want to emit a Chorus MIDI controller (MIDI CC 93) at the specified value on this Generator's MIDI line. Leave this value at the default of "-1" if you don't want the Generator to emit any information for this MIDI controller.

Damper Release

If you are using Damper/Hold (64), then you will find that your notes can start building-up and never decay! In which case, set the Damper Release parameter to "Yes", which tells the Generator to momentarily release the damper just before the end of every bar. This prevents build-up of notes and generally sounds wonderful.

Portamento (65)

Set this value to other than the default of "-1", if you want to emit a Portamento MIDI controller (MIDI CC 65) at the specified value on this Generator's MIDI line. Leave this value at the default of "-1" if you don't want the Generator to emit any information for this MIDI controller.

MIDI Channel Sharing

The default value of "Yes" means that this Generator can share its MIDI channel with other Generators. This is only considered if you have defined the MIDI Channel parameter for a Generator to be 0.

Generator - Micro Controllers ^

Overview

The Generator Microcontroller parameters (there are two - User C1 and User C2) allow you to define very powerful Microcontrollers to be associated with your Generator.

Microcontrollers are very powerful and you can think of them as built-in, highly configurable MIDI event generators. They can either synchronise to the tempo of your Cell, or you can let them run free-floating. Experiment with them – they can do a huge amount to make your music interesting and dynamic.

Tip: if you want to synchronise your Microcontroller to the time-base, so that your MIDI controller is synchronised to bar boundaries in your music, you'll need to use the Beat Cycle Length parameter.

Parameter Group - Micro Controllers

MIDI CC
Mode
Minimnum
Range
Change
Change Range
Update
Update Range
Update Units
Beat Cycle Length
Phase Shift %

MIDI CC ^

This tells your Generator which MIDI controller (also referred to as the MIDI CC) to emit for this microcontroller, e.g. 11. When the Microcontroller is active, the Generator will emit values for this MIDI controller that change at various times, with behaviour that you define using the various parameters in this Parameter.

Mode ^

The Mode defines the shape of the waveform that the Generator will use to shape this waveform.

The Mode may be one of the following values:

- *-1 – Off*
  The microcontroller is off. This is the default value.
- *0 - Random Drift*
  The microcontroller will drift between the Minimum and Minimum plus Range, changing at times specified by the Update and Update Range parameters, by an amount between the Change and Change plus Change Range parameters.
- *1 - LFO (Min-Max-Min)*

A triangular waveform, that starts at the minimum value, works up to the maximum value, and works back to the minimum value.

- *2 - LFO (Max-Min-Max)*
  A triangular waveform, that starts at the maximum value, works down to the minimum value, and works back to the maximum value.
- *3 - Sawtooth (Min-Max)*
  A sawtooth waveform, that starts at the minimum value, works up to the maximum value, and then starts again from the minimum value.
- *4 - Sawtooth (Max-Min)*
  A sawtooth waveform, that starts at the maximum value, works down to the minimum value, and then starts again from the maximum value.

Minimum ^

Defines the minimum value that may be emitted by the Microcontroller.

Range ^

The microcontroller will emit a value between the Minimum and Minimum plus Range values.

So for example, if you define Minimum to be 20, and Range to be 100, the value that is emitted will be in the range 20 to 120 inclusive.

Change ^

Defines the amount by which the microcontroller will change, every time it is allowed to change. Typically set to a value of 1. If this value is set to 0, the Microcontroller will change only if the Change Range is greater than or equal to 1.

Change Range ^

Defines the upper limit to the amount by which the microcontroller will change, every time it is allowed to change. Typically set to a value of 1. If this value is set to 0, the Microcontroller will change only if the Change Range is greater than or equal to 1.

For example, if you define Change to be 1, and Change Range to be 3, the value that is emitted will vary by a value between 1 and (3+1)=4 each time.

Update ^

Defines the minimum time in milliseconds between changes in the emitted Microcontroller value. The system might not be able to emit changes as quickly as you want, if you set a very small value! If you don't want changes to happen very often, then use a large value.

Ignored if Beat Cycle Length is non-zero.

Update Range ^

Defines the upper limit in the time in milliseconds between changes in the emitted Microcontroller value. Use this parameter to apply some uncertainty in when the changes will occur.

For example, if you define Update to be 1000, and Update Range to be 500, the value that is emitted will change every 1000 to 1500 milliseconds (or in other words, every 1 to 1.5 seconds).

Ignored if Beat Cycle Length is non-zero.

Update Units ^

You define the Unit of Measure by which the Update and Update Range parameters are interpreted. This may be one of the following values:

- *Seconds (thousandths of a)*
  The Update and Update Range are interpreted as being in thousandths of a second (i.e. Milliseconds). So, a Update value of 1000 means one second.
- *Full seconds*
  The Update and Update Range are interpreted as being in seconds. So, a Update value of 10 means ten seconds.

Beat Cycle Length ^

This parameter is critical for generating effects which synchronise with the bar timing of your Generator. If you want to achieve an effect like a filter-sweep that synchronises to your bar boundary, then this is the parameter to use.

Here are some of the values you could use.

Note in the IME a Beat is defined as being one crotchet; you get 4 beats in a bar of 4:4 music. So, a Duration value of 60 means one beat. A Duration value of 30 means a quaver. A Duration value of 20 means a triplet. A Duration value of 15 means a semi-quaver. A Duration value of 240 means 4 beats (which is a full bar if the Cell Meter is 4:4).

Phase Shift % ^

Use this parameter if you want to start the microcontroller from a start-point other than at the very start of its cycle.

Generator - Micro Delay

Overview

The Generator Micro Note Delay parameters provide fine variation in the times of Note events generated by a Generator. This can be used to give a Generator more “human” feel.

Parameter Group - Micro Note Delay

Delay Range

Delay Offset

Delay Change

Delay Range

The maximum amount of delay generated by micro note delay changes, that may be applied to note events. Zero means off (which is the default).

Delay Change

The amount of change in the micro delay that is applied by Wotja between note on/off events. The value drifts between zero (off) and the Delay Range, changing by plus or minus the Delay Change value each time.

Delay Offset

Fixed amount of offset note delay to apply, used only when the Micro Note Delay controller is in use. The default value is zero.

Generator - Micro Pitch

Overview

The Generator Micro Pitch parameters provide fine variation through use of the MIDI Pitch Wheel controller.

Tip: This is not normally used on MIDI line 10, which is the drum/percussion line!

Parameter Group - Micro Pitch

Bend Sensitivity
Pitch Bend Offset
Pitch Range
Pitch Change
Pitch Update
Pitch Update Range

Bend Sensitivity

A value from 0 to 24, meaning how many semitones are controlled by the full available range of Micro Pitch parameters. The default value is 2, which represents two semitones.

Pitch Bend Offset

Fixed amount of pitch-bend to apply on this MIDI line, used to tune/de-tune an instrument.

The default value is zero, which means to apply no offset pitch bend.

From -8192 to +8191; which covers a range of pitch bend defined by the Bend Sensitivity parameter.

Pitch Range

The maximum amount of micro pitch change that can be applied. Zero means off (which is the default). The maximum value allowed is 8191. The value chosen is added to the pitch bend offset.

Pitch Change

The amount of change in Micro Pitch that is applied by Wotja between “update” periods. The value drifts between zero (off) and the Pitch Range, changing by plus or minus the Pitch Change value each time.

Pitch Update

The time in milliseconds between updates to the pitch controller. The actual value chosen is selected randomly each time, to be a value somewhere between Pitch Update and Update Range.

Pitch Update Range

The upper limit of time between updates to the pitch controller. The actual value chosen is betwen Pitch Update and Update Range.

Generator - Note to MIDI CC

Overview

Normally, Generators compose and emit MIDI note events. The Generator Note to MIDI CC Mapping parameters allow you to tell a Generator to emit MIDI controller data instead of MIDI note events.

Why would you want to do this? Well, it lets you use a Generator as a very powerful generative MIDI event generator with a huge range of potential applications.

Parameter Group - Note to MIDI CC

CC for Note On?
Note On CC
CC for Velocity?
Velocity CC
CC for Note Off?
Note Off CC

CC for Note On?

If you want this Generator to emit a MIDI CC instead of note on/off events, set this parameter to Yes.

Note On CC

If you have set CC for Note On? to Yes, then instead of emitting a note on event, the Generator will emit the specified MIDI CC, with a value equal to the composed pitch.

CC For Velocity?

If you want this Generator to emit a MIDI CC proportionate to the Velocity of the composed note (in addition to any controller defined for Note On CC), then set this parameter to Yes.

Velocity CC

If you have set CC for Velocity? to Yes, then the Generator will (in addition to the Note On CC value) emit the specified MIDI CC, with a value equal to the composed velocity.

CC for Note Off?

If you want this Generator to emit a MIDI CC when a note off occurs, set this parameter to Yes. This applies only if CC for Note On? Is set to Yes.

Note On CC

If you have set CC for Note Off? to Yes, then instead of emitting a note off event, the Generator will emit the specified MIDI CC, with a value equal to the composed pitch of the stopped note.

Generator - Envelopes ^

Overview

See also the Envelope Editor.

Generator Envelopes are supported for a number of parameters. Envelopes work in the same way for all of these, so they are all grouped here.

Each envelope is a collection of up to 100 data points. A Cell starts with the value at the left side of the envelope, and as it progresses, eventually ends up with the value from the far right of the envelope.

You can draw direct on to the envelope with your mouse.

Alternatively, you may use one of the various powerful envelope editing tools that we have made available to you.

To use the envelope editing tools:
- Right-click (win) or ctrl-click (mac) on the envelope tool
- Select the option you want. e.g. random, curve up etc.
- Select the range using the mouse...
- Then: either press space or enter, or select pop-up envelope tool to apply to the selected range.
- Select freehand mode to return to the normal click-to-paint mode.

Parameter Group - Envelopes

Velocity
Velocity Range
Velocity Change
Velocity Change Range
User Envelope 1 (Volume)
Micro User (Envelope 1)
User Envelope 2 (Pan)

Velocity Envelope ^

This allows you to define how the Velocity level is changed automatically for your Generator throughout playback of the Cell. The velocity defines relatively how loud each note is.

The actual Velocity value used at any point in the Cell is calculated as being a value somewhere in the range from Velocity, to Velocity plus the value of the Velocity Range envelope.

Velocity Range Envelope ^

The value in this envelope is added to the value in the Velocity Envelope.

Velocity Change Envelope ^

The velocity for any composed note can change by a value between Velocity Change Envelope value (VCE) and the VCE plus the Velocity Change Range Envelope value.

Velocity Change Range Envelope ^

The value in this envelope is added to the value in the Velocity Change Envelope.

User Envelope 1 (Volume) ^

This allows you to define an envelope that is used to emit a MIDI CC of your choice. The default value for this MIDI CC is 7, which is the Volume controller.

- **MIDI CC**

  Use this to define the MIDI CC that you want to be emitted by this envelope. The default value is 7, which is the Volume controller.

- **Enabled?**

  Use this to turn your envelope on or off.

- **Envelope**

Micro User (Envelope 1) ^

This provides fine variation in the values generated by the User Envelope 1. Any value generated by this micro controller is added to the Micro User Envelope 1 value, to give fine variation in any such envelope.

- **Range**

  The maximum amount of micro change in the User Envelope 1 that can be applied. Zero means off (which is the default).

- **Change**

  The amount of micro change in the User Envelope that is applied by Wotja between “update” periods. The value drifts between zero (off) and the Range, changing by plus or minus the Change value each time.

- **Update**

  The time in milliseconds between updates to the Micro User Envelope value. The actual value chosen is selected randomly each time, to be a value somewhere between Update and Update Range.

- **Update Range**

  The upper limit of time between updates to the Micro User Envelope value. The actual value chosen is betwen Update and Update Range.

User Envelope 2 (Pan) ^

This allows you to define an envelope that is used to emit a MIDI CC of your choice. The default value for this MIDI CC is 10, which is the Pan controller.

- **MIDI CC**

  Use this to define the MIDI CC that you want to be emitted by this envelope. The default value is 10, which is the Pan controller.

- **Enabled?**

  Use this to turn your envelope on or off.

- **Envelope**