



NATIVE INSTRUMENTS
SOFTWARE SYNTHESIS

REAKTOR 5


オペレーションマニュアル



このマニュアルに記載した事項は、予告なしに変更される場合があります。また、内容についてNative Instruments Software Synthesis GmbHが責任を負うと表明するものではありません。このマニュアルに述べるソフトウェアの使用にあたっては、利用許諾契約を結ぶ必要があります。ソフトウェアを他の媒体に複製することはできません。このマニュアルの一部または全部をコピー、転載し、または伝送、録音、録画するためには、目的のいかんを問わず、Native Instruments Software Synthesis GmbHから事前に書面による許可を得る必要があります。このマニュアルに記載した製品名や会社名は、それぞれの所有者の商標または登録商標です。

マニュアルの執筆者: Rick Scott, Marius Wilhelmi, Len Sasso, Stephan Schmitt, Erik Wiegand, James Walker-Hall, Julian Ringel。
制作協力: Henri Hagenow, Dan Santucci, Felski。

© Native Instruments Software Synthesis GmbH, 2005. All rights reserved. REAKTOR, REAKTOR 5, REAKTOR 5 COREはNative Instruments Software Synthesis の商標です。

NATIVE INSTRUMENTS Software Synthesis 
info@native-instruments.com www.native-instruments.com

目次

1. はじめに	1
1.1. REAKTOR とは?	1
1.2. REAKTOR 5 で新たに加わった機能と変更点	1
1.3. イベント初期設定	2
1.4. REAKTOR Core 技術	2
1.5. 新しい基本モジュール	3
1.6. 機能や外観を変更した基本モジュール	4
1.7. 新しい機能	4
1.8. 機能変更	5
1.9. 廃止した機能、実行手順を変更した機能	6
1.10. REAKTOR 3 で作成したアンサンプルの取り扱い	7
2. 製品の認証手続き	8
2.1. 認証手続きの概要	8
2.2. 認証手続きの手順	9
2.3. 手順 A: 稼動コンピューターが直接インターネット接続されている場合	9
2.4. 手順 B: 別のコンピューターがインターネット接続されている場合	12
2.5. 手順 C: インターネットにつながっていない場合	14
2.6. 登録に関する技術サポート	16
3. Windows XP 版のインストール手順	18
3.1. 動作条件と推奨システム環境	18
3.2. ソフトウェアのインストール手順	18
3.3. VST プラグ・インのインストール	19
3.4. DXi 2 プラグ・インの設定	19
3.5. RTAS プラグ・インのインストール	20
4. MacOS X 版のインストール手順	21
4.1. 動作条件と推奨システム環境	21
4.2. ソフトウェアのインストール手順	21
4.3. MacOS Audio Unit プラグ・インのインストール	22
4.4. RTAS プラグ・インのインストール	22

5. オーディオ・インターフェイス	23
6. スタンドアローン版の REAKTOR 5.....	27
6.1. Soundcard タブ (オーディオ・インターフェイス).....	27
6.2. Routing タブ.....	29
6.3. MIDI タブ.....	31
7. プラグ・イン版の REAKTOR 5.....	32
7.1. オートメーション ID の編集.....	33
7.2. ファイル管理、特に自動保存の機能について	33
7.3. VST 2.0 プラグ・イン	37
7.4. Audio Units プラグ・イン	40
7.5. DXi 2 プラグ・イン	44
7.6. Pro Tools 6.x での REAKTOR RTAS の使い方	45
8. OSC(Open Sound Control)	47
8.1. 適用分野.....	47
8.2. OSC システムの設定.....	48
9. REAKTOR 最初の一步.....	52
9.1. 作成済みインストゥルメントの演奏.....	52
9.2. 基本的なシンセサイザーの構築.....	63
9.3. 独自のストラクチャーの構築.....	77
10. REAKTOR の基本操作	83
10.1. マウス.....	83
10.2. コンテキスト・メニュー.....	84
10.3. キーコマンド.....	84
10.4. アンサンブル制御パネル、ストラクチャー・ウィンドウ	84
11. メニュー・コマンド	86
11.1. File メニュー.....	86
11.2. Edit メニュー.....	88
11.3. Settings メニュー.....	90
11.4. System メニュー.....	92
11.5. View メニュー.....	99
11.6. ?メニュー	105

12. REAKTOR のツールバー	107
12.1. 主ツールバー	107
12.2. アンサンブル制御パネルのツールバー	109
12.3. ストラクチャー・ウィンドウのツールバー	110
13. ブラウザー	112
13.1. ファイルの操作	113
13.2. オーディオ・ファイルの試聴	115
14. アンサンブル	117
14.1. アンサンブル・ストラクチャー・ウィンドウ	119
14.2. アンサンブル制御パネル・ウィンドウ	120
14.3. アンサンブルのプロパティ画面	121
15. インストゥルメント	129
15.1. インストゥルメントをアンサンブルに追加する手順	129
15.2. ポート	130
15.3. コンテキスト・メニュー	130
15.4. インストゥルメント・ヘッダー	131
15.5. インストゥルメントのプロパティ画面	133
16. 基本マクロ	145
16.1. 「基本マクロ」の目的	145
16.2. ストラクチャーにマクロを追加する手順	146
16.3. ポート	147
16.4. コンテキスト・メニュー	147
16.5. マクロのプロパティ画面	148
17. 基本ストラクチャー	153
17.1. 「基本ストラクチャー」の役割	153
17.2. モジュール	154
17.3. 信号源モジュール	159
17.4. スイッチ	161
17.5. 入出力端子	161
17.6. 結線	162
17.7. 信号処理	165
17.8. コンテキスト・メニュー	169

18. 制御パネルの編集	171
18.1. 制御パネル	171
18.2. 制御パネルのつまみ類	172
18.3. 各種のつまみ類	172
18.4. つまみ類のスキン	180
18.5. つまみ類に関する Connection プロパティ	185
18.6. 制御パネルの編集	188
19. パネル操作	189
19.1. マウスによるつまみ類の操作	189
19.2. キーボードによるつまみ類の操作	193
19.3. MIDI 信号によるつまみ類の制御	194
19.4. MIDI 出力	195
19.5. 独自のパネルの定義	195
20. スナップショット	201
21. サンプリングと再合成	211
21.1. サンプルの管理	211
21.2. サンプル・マップ	214
21.3. サンプル・マップ・エディター	217
21.4. Akai ファイルのインポート	226
22. テーブル型のモジュール	229
22.1. プロパティ	229
22.2. コンテキスト・メニュー	237
22.3. その他の機能	240
23. 「Classic Modular」 マクロ・コレクション	241
23.1. 表示系マクロ	242
23.2. MIDI 処理系マクロ	243
23.3. ミキサー / アンプ系マクロ	244
23.4. 発振器系マクロ	247
23.5. サンプラー系マクロ	248
23.6. シーケンサー系マクロ	250
23.7. LFO、エンベロープ系マクロ	256
23.8. フィルター系マクロ	258
23.9. ディレイ系マクロ	260

23.10. 歪み系マクロ	261
23.11. イベント処理系マクロ	263
モジュール・リファレンス	264
パネル系モジュール	266
Fader (フェーダー)	266
Knob (ノブ)	268
Button (ボタン)	269
List (リスト / メニュー)	270
Switch (スイッチ)	271
Lamp (ランプ)	272
Level Lamp (レベル・ランプ)	273
RGB Lamp (RGB ランプ)	273
Meter (メーター)	274
LevelMeter (レベル・メーター)	275
Picture (ピクチャー)	275
Multi Picture (マルチ・ピクチャー)	275
Text (テキスト)	276
Multi Text (マルチ・テキスト)	276
XY (2次元グラフ / コントローラー)	277
Scope (スコープ)	278
Multi Display、Poly Display	279
Mouse Area (マウス領域)	281
Stacked Macro (マクロの切り替え表示枠)	283
MIDI 入力系モジュール	284
Note Pitch (ノート・ピッチ)	284
Pitchbend (ピッチベンド)	284
Gate (ゲート)	285
Single Trig. Gate (シングル・トリガー型ゲート)	285
Sel. Note Gate (ノート番号選択型ゲート)	285
On Velocity (オン・ベロシティ)	285
Off Velocity (オフ・ベロシティ)	285
Controller (コントローラー)	286
Ch. Aftertouch (チャンネル・アフタータッチ)	286
Poly Aftertouch (ポリ・アフタータッチ)	287

Sel. Poly AT (ノート番号選択型ポリ・アフタータッチ).....	287
Program Change (プログラム・チェンジ).....	287
Start/Stop (開始 / 停止).....	288
1/96 Clock (96 分音符クロック).....	288
Sync Clock (同期クロック).....	288
Song Pos (ソング位置).....	289
Channel Message (チャンネル・メッセージ).....	289
MIDI 出力系モジュール	291
Note Pitch/Gate (ノート・ピッチ / ゲート).....	291
Pitchbend (ピッチベンド).....	291
Controller (コントローラー).....	291
Ch. Aftertouch (チャンネル・アフタータッチ).....	292
Poly Aftertouch (ポリ・アフタータッチ).....	292
Sel. Poly AT (ノート番号選択型ポリ・アフタータッチ).....	292
Program Change (プログラム・チェンジ).....	293
Start/Stop (開始 / 停止).....	293
1/96 Clock (96 分音符クロック).....	293
Song Pos (ソング位置).....	293
Channel Message (チャンネル・メッセージ).....	294
数学演算系モジュール	295
Constant (定数値).....	295
Add (加法).....	295
Subtract (減法).....	296
Invert (符号反転、-x).....	296
Multiply (乗法).....	296
$a * b + c$	297
Reciprocal (逆数、 $1/x$).....	297
Divide (除法).....	297
Modulo (剰余).....	298
Rectifier (整流、絶対値).....	298
Rectify/Sign (整流、絶対値 / 符号).....	298
Compare (比較).....	299
Compare/Equal (比較).....	299
Quantize (量子化 / クォンタイズ).....	300

Expon. (A) (指数演算 - 振幅)	300
Expon. (F) (指数演算 - 周波数)	300
Log (A) (対数演算 - 振幅)	301
Log (F) (対数演算 - 周波数)	301
Power (冪、 x^y)	301
Square Root (平方根)	302
1 / Square Root (平方根の逆数)	302
Sine (正弦)	302
Sine/Cos (正弦 / 余弦)	303
Arcsin (逆正弦)	303
Arccos (逆余弦)	303
Arctan (逆正接)	304
信号配送系モジュール	305
Selector/Scanner (セレクター / スキャナー)	305
Relay (リレー)	306
Crossfade (クロスフェード)	306
Distributor/Panner (信号分配器 / パン)	306
Stereo Pan (ステレオ・パン)	307
Amp/Mixer (アンプ / ミキサー)	308
Stereo Amp/Stereo Mixer (ステレオ・アンプ / ミキサー)	308
発振器系モジュール	309
Sawtooth (鋸波)	309
Saw FM (鋸波、周波数変調機能つき)	309
Saw Sync (鋸波、位相同期機能つき)	310
Saw Pulse (鋸波パルス)	311
Bi-Saw (両極鋸波)	311
Triangle (三角波)	312
Tri FM (三角波、周波数変調機能つき)	312
Tri Sync (三角波、位相同期機能つき)	313
Tri/Par Symm (三角波 / 放物波、対称性の調整機能つき)	313
Parabol (放物波)	314
Par FM (放物波、周波数変調機能つき)	314
Par Sync (放物波、位相同期機能つき)	315
Par PWM (放物波、パルス幅変調機能つき)	315

Sine (正弦波)	316
Sine FM (正弦波、周波数変調機能つき)	316
Sine Sync (正弦波、位相同期機能つき)	317
Multi-Sine (正弦波の加算合成)	318
Pulse (パルス波)	318
Pulse FM (パルス波、周波数変調機能つき)	319
Pulse Sync (パルス波、位相同期機能つき)	320
Pulse 1-ramp (片台形波)	321
Pulse 2-ramp (両台形波)	321
Bi-Pulse (両極パルス波)	322
Impulse (衝撃波、インパルス)	323
Impulse FM (衝撃波、周波数変調機能つき)	323
Impulse Sync (衝撃波、位相同期機能つき)	324
Multi-Step (階段波)	325
4-Step (4 段階の階段波)	325
5-Step (5 段階の階段波)	325
6-Step (6 段階の階段波)	325
8-Step (8 段階の階段波)	325
Multi-Ramp (折れ線波)	326
4-Ramp (4 段階の折れ線波)	326
5-Ramp (5 段階の折れ線波)	326
6-Ramp (6 段階の折れ線波)	326
8-Ramp (8 段階の折れ線波)	326
Ramp (ランプ波)	327
Clock Oscillator (クロック信号発振器)	327
Noise (雑音発生器)	328
Random (乱数信号)	328
Geiger (ガイガー・カウンター)	329
Audio Table (オーディオ・テーブル)	329
サンプラー系モジュール	331
Sampler (サンプラー)	332
Sampler FM (サンプラー、周波数変調機能つき)	333
Sampler Loop (サンプラー・ループ)	334
Grain Resynth (グレイン・リシンス)	336

Grain Pitch Former (グレイン・ピッチ・フォーマー)	340
Grain Cloud (グレイン・クラウド)	344
Beat Loop (ビート・ループ)	346
Sample Lookup (サンプル・ルックアップ)	348
シーケンサー系モジュール	349
Sequencer (シーケンサー)	349
6-Step (6 段ステップ・シーケンサー)	349
8-Step (8 段ステップ・シーケンサー)	350
12-Step (12 段ステップ・シーケンサー)	350
16-Step (16 段ステップ・シーケンサー)	350
Multiplex 16 (16 段マルチプレクサー)	351
LFO/ エンベロープ系モジュール	352
LFO (低周波発振器)	352
Slow Random (スロー・ランダム)	353
H-Env (H- エンベロープ)	353
HR-Env (HR- エンベロープ)	354
D-Env (D- エンベロープ)	354
DR-Env (DR- エンベロープ)	355
DSR-Env (DSR- エンベロープ)	355
DBDR-Env (DBDR- エンベロープ)	356
DBDSR-Env (DBDSR- エンベロープ)	357
AD-Env (AD- エンベロープ)	358
AR-Env (AR- エンベロープ)	358
ADR-Env (ADR- エンベロープ)	359
ADSR-Env (ADSR- エンベロープ)	360
ADBDR-Env (ADBDR- エンベロープ)	361
ADBDSR-Env (ADBDSR- エンベロープ)	362
AHDSR-Env (AHDSR- エンベロープ)	363
AHDBDR-Env (AHDBDR- エンベロープ)	364
4-Ramp (4- ランプ)	365
5-Ramp (5- ランプ)	366
6-Ramp (6- ランプ)	366
フィルター系モジュール	367
HP/LP 1-Pole (ハイ / ロー・パス、1P)	367

HP/LP 1-Pole FM (ハイ / ロー・パス、1P、FM).....	368
Allpass 1-Pole (オール・パス、1P).....	368
Multi 2-Pole (マルチ、2P).....	369
Multi 2-Pole FM (マルチ、2P、FM).....	369
Multi/Notch 2-Pole (マルチ / ノッチ、2P).....	370
Multi/Notch 2-Pole FM (マルチ / ノッチ、2P、FM).....	371
Multi/LP 4-Pole (マルチ / ロー・パス、4P).....	372
Multi/LP 4-Pole FM (マルチ / ロー・パス、4 ポール、FM).....	373
Multi/HP 4-Pole (マルチ / ハイ・パス、4P).....	374
Multi/HP 4-Pole FM (マルチ / ハイ・パス、4P、FM).....	375
Pro-52 Filter (Pro-52 フィルター).....	376
Ladder Filter (ラダー・フィルター).....	377
Ladder Filter FM (ラダー・フィルター、FM).....	377
Peak EQ (ピーク・イコライザー).....	378
Peak EQ FM (ピーク・イコライザー、FM).....	379
High Shelf EQ (ハイ・シェルフ・イコライザー).....	379
High Shelf EQ FM (ハイ・シェルフ・イコライザー、FM).....	380
Low Shelf EQ (ロー・シェルフ・イコライザー).....	380
Low Shelf EQ FM (ロー・シェルフ・イコライザー、FM).....	381
Differentiator (微分器、ディファレンシエーター).....	382
Integrator (積分器、インテグレーター).....	382
ディレイ系モジュール.....	383
Single Delay (シングル・ディレイ).....	383
Multi-Tap Delay (マルチ・タップ・ディレイ).....	384
Diffuser Delay (ディフューザー・ディレイ).....	384
Grain Delay (グ레인・ディレイ).....	385
Grain Cloud Delay (グ레인・クラウド・ディレイ).....	387
Unit Delay (ユニット・ディレイ).....	389
歪み系 (ディストーション系) モジュール.....	390
Saturator (サチュレーター).....	390
Saturator 2 (サチュレーター 2).....	390
Clipper (クリッパー).....	391
Mod. Clipper (モジュレーション・クリッパー).....	391
Mirror 1 Level (1 レベル・ミラー).....	392

Mirror 2 Levels (2 レベル・ミラー).....	392
Chopper (チョッパー).....	392
Shaper 1 BP (シェイパー、1 BP).....	393
Shaper 2 BP (シェイパー、2 BP).....	393
Shaper 3 BP (シェイパー、3 BP).....	394
Shaper Parabolic (シェイパー、2 次の特性曲線).....	394
Shaper Cubic (シェイパー、3 次の特性曲線).....	395
Slew Limiter (スルー・リミッター).....	395
Peak Detector (ピーク・デテクター).....	396
Sample & Hold (サンプル & ホールド).....	396
Frequency Divider (分周器).....	397
イベント処理系モジュール.....	398
Accumulator (累算器、アキュムレーター).....	398
Counter (計数器、カウンター).....	398
Randomizer (乱数生成器).....	399
Frequency Divider (分周器).....	399
Ctrl. Shaper 1 BP (制御信号用シェイパー、1 BP).....	400
Ctrl. Shaper 2 BP (制御信号用シェイパー、2 BP).....	400
Ctrl. Shaper 3 BP (制御信号用シェイパー、3 BP).....	401
Logic AND (論理積).....	401
Logic OR (論理和).....	401
Logic EXOR (排他的論理和).....	402
Logic NOT (論理否定).....	402
Order (順序制御).....	402
Iteration (イテレーション、反復).....	403
Separator (セパレーター、条件分岐).....	403
Value (値変更).....	404
Merge (マージ、併合).....	404
Step Filter (ステップ・フィルター).....	404
Router M->1 (ルーター M → 1).....	405
Router 1,2 (ルーター 1/2).....	405
Router 1->M (ルーター 1 → M).....	406
Timer (タイマー).....	406
Hold (ホールド).....	406

Event Table (イベント・テーブル)	407
補助系モジュール	409
Tapedeck 1-Ch (1- チャンネル・テープデッキ)	409
Tapedeck 2-Ch (2- チャンネル・テープデッキ)	411
Audio Voice Combiner (オーディオ声部合成)	412
Event V.C. - All (イベント声部合成 - 全声部)	412
Event V.C. - Max (イベント声部合成 - 最大値の声部)	412
Event V.C. - Min (イベント声部合成 - 最小値の声部)	413
A to E (オーディオ→イベント変換)	413
A to E (Trig) (オーディオ→イベント変換、トリガーつき)	413
A to E (Perm) (オーディオ→イベント変換、一定周期)	414
A to Gate (オーディオ→ゲート変換)	414
To Voice (特定声部への転送)	414
From Voice (特定声部からの転送)	415
Voice Shift (声部番号のシフト)	415
Audio Smoother (オーディオ・スモーカー)	416
Event Smoother (イベント・スモーカー)	416
Master Tune/Level (マスター・チューン / レベル)	417
Tempo Info (テンポ情報)	417
Voice Info (声部情報)	417
Tuning Info (調律情報)	418
System Info (システム情報)	418
Note Range Info (音域情報)	419
MIDI Channel Info (MIDI チャンネル情報)	419
Snapshot (スナップショット操作)	420
Set Random (乱数の種)	422
Unison Spread (ユニゾン・スプレッド)	422
Snap Value (スナップショットの付値)	423
Snap Value Array (スナップショットの付値 - 配列)	423
入出力端子系モジュール	425
In Port (入力ポート)	425
Out Port (出力ポート)	425
Send (信号送信)	425
Receive (信号受信)	425

IC Send (アンサンブル内送信)	427
IC Receive (アンサンブル内受信)	427
OSC Send (OSC 送信)	427
OSC Receive (OSC 受信)	428
付録 音符に対応するコンピューター・キーボード上のキー	429

1. はじめに

1.1. REAKTOR とは？

REAKTOR があれば、ごく普通のコンピューターが、強力なシンセサイザー兼オーディオ処理システムに生まれ変わります。REAKTOR は完全モジュール構成であり、どんな音声合成 / エフェクト処理システムでも、思うがままに構築できます。比較的単純なアナログ・シンセサイザーから大規模モジュラー・システム、基本的なサンプル・プレイヤーから、波形を分割して並べ替えるグラニューラー方式を採り入れた高度なシステム、さらには単純なディレイ・ラインから多機能リバーブ装置まで、創造力は無限に広がるでしょう。

独自のインストゥルメントやエフェクトを作ることが主目的でなくても、REAKTOR には汲めども尽きぬ魅力が詰まっています。数百種類に及ぶさまざまなインストゥルメントやエフェクトが、ライブラリーとして用意されているのです。単純な FM シンセサイザーはもちろん、再生速度とピッチを独立に制御できるサンプル・プレイヤーや、原形をとどめぬほどにまでオーディオ・ファイルを書き替えてしまうマルチ・エフェクト・ボックスまでもが揃っています。なによりも REAKTOR ライブラリーには、思いついたら即座に曲作りに取りかけられる、という長所があります。

もちろんライブラリーをそのまま使ったのでは思った通りのことができない場合もありますが、モジュール構成はどのようにでも修正が可能です。隠れている機能や、変更できないよう保護された箇所はありません。活発なユーザーも多く、それぞれが作り上げたインストゥルメントやエフェクトが、オンライン・ライブラリーとして続々と蓄積されています。REAKTOR の使い方はさまざまです。まずはあらかじめ用意されているアンサンプル・ライブラリーを使い、次に自分なりの修正を試みましょう。こうして経験を積んだら、いよいよ独自のインストゥルメントやエフェクトの構築に取り組んでください。とにかく始めることが大切です。

1.2. REAKTOR 5 で新たに加わった機能と変更点

REAKTOR 5 では、従来にも増して柔軟かつ強力な音声データ処理能力が加わりました。以下、新機能や変更点の概要を紹介します。

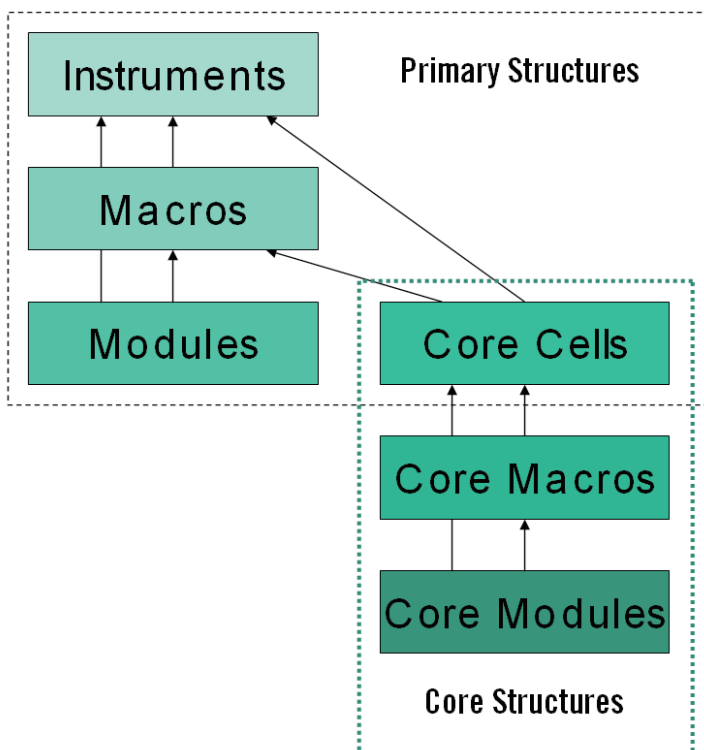
1.3. イベント初期設定

イベント入力の初期設定に新しい方式が加わりました。これは REAKTOR 4 との互換モードをオフにした場合に有効です。将来の互換性を確保するためにも、オフにしておくようお勧めします。

1.4. REAKTOR Core 技術

REAKTOR 5 で最も大きく変わった点がこの REAKTOR Core 技術です。機能を基本レベルとコア・レベルの 2 階層に分けて管理する方式です。

基本レベルのストラクチャーは、REAKTOR 4 と同様に、インストールメント、マクロ、モジュールで構成されています。なお、インストールメントの中でも、階層構成の最上位に位置するものを特に「アンサンブル」と呼びます。



一方、新たに加わったコア・レベルは、コア・セル、コア・マクロ、コア・モジュールから成ります。コア・セル (*.rcc ファイル) は、コア・マクロやコア・モジュールを組み合わせたもので、基本レベルとコア・レベルの橋渡しをする役割があります。基本レベルのインストゥルメントは、マクロやモジュールを組み合わせで組み立てますが、これと同じ関係ということになります。

マクロは、基本レベル、コア・レベルとも、外部ファイル (それぞれ *.mdl, *.rcm) に保存するようになっています。一方モジュールは、REAKTOR プログラム本体に一体化されているので、「組み込みモジュール」と称することもあります。

コア・セルや、これに含まれるコア・マクロ、コア・モジュールは、実行のしくみや信号の伝播方式が基本レベルとは違ってしています。これにより、REAKTOR 4 では不可能だった、洗練された信号処理ストラクチャーを作成できるようになりました。詳しくは別冊「REAKTOR CORE チュートリアル / オペレーション・マニュアル」を参照してください。

1.5. 新しい基本モジュール

オーディオ信号の生成 / 処理に関する新しいモジュールはありません。低レベルの DSP 処理については、基本レベルではなくコア・レベルで扱うようになったので、今後はこちらのライブラリーを充実させていく予定です。

新しい基本モジュールは主として、ユーザー・インターフェイスやデータ保存、声部の切り替え、MIDI 入出力、内部接続に関するものです。具体的には次のようなものがあります。

- ・ **Mouse Area** (パネル系): 他のモジュール (**Multi Display**、**Poly Display** など) を、マウスで操作 (クリック / ドラッグ / カーソル位置移動など) できるようにする働きがあります。
- ・ **Multi Display**、**Poly Display** (パネル系): 矩形、静止画、動画などの画像オブジェクトを、複数まとめて生成、処理できます。
- ・ **Stacked Macro**、**Panel Index** (パネル系): インストゥルメント制御パネル内のある表示領域を複数のマクロで共有し、切り替えながら表示できるようにする働きがあります。
- ・ **Channel Message** (MIDI 入力系)、**Channel Message** (MIDI 出力系): 各種の MIDI チャンネル・メッセージを、外部 MIDI 機器 (キー

ボード、シーケンサー、ファイルなど)や内蔵インストゥルメントとの間で送受信します。

- ・ **Voice Shift** (補助系): ある番号の声部 (例: 1、2 番) を別の番号 (例: 3、4 番) に割り当て直して出力します。
- ・ **Snap Value Array** (補助系): 一連の浮動小数点数の配列を、スナップショットと組にして保存し、必要に応じて再び読み込みます。
- ・ **IC Send、IC Receive** (入出力端子系): アンサンプル内のモジュール間で、単声のイベント信号を送受信します。「IC」は「Internal Connection」を表します。
各モジュールについて詳しくは、「基本モジュール・リファレンス」を参照してください。

1.6. 機能や外観を変更した基本モジュール

次の基本モジュールは、機能や外観が変わりました。

- ・ **Invert、Rectify** (数学演算系)、**Merge、Order、Value、Logic AND、Logic OR、Logic XOR、Logic NOT** (イベント処理系): ストラクチャー・アイコンが変わりました。
- ・ **Meter、Lamp、Multi Picture、Multi Text** (パネル系)、**MIDI In Controller、MIDI Out Controller**: アンサンプル内の接続を表す **Internal Connections** プロパティがなくなりました。代わりに **IC Send、IC Receive** モジュールを使ってください。
- ・ **Snap Value** (補助系): 単声だけでなく、多声モードでも使えるようになりました。
- ・ パネル系モジュール: つまみ類の配置やラベルは、制御パネルのロックを解除すれば直接編集できるようになりました。ロックした状態では、値を設定できます。また、つまみ類のほとんどについて、見た目を好きなように変更できる「スキン」機能がつけました。インストゥルメントや基本マクロの背景画像も変更可能です。詳しくは「基本モジュール・リファレンス」を参照してください。

1.7. 新しい機能

次のような機能を追加しました。

- ・ パネルセット: 「スクリーンセット」の機能を拡張したものです。
- ・ ストラクチャーの「しおり」機能: しおりを設定しておけば、他のストラクチャーを操作中であっても即座にその画面に切り替え可能です。

- ・ インストゥルメントの声部割り当て設定のロック: **VOICES**、**MAX UNISON V**、**MIN UNISON V** の各設定を、誤って変更してしまわないようロックできます。ロックのオン / オフは **Lock Voices** プロパティで切り替えます。
- ・ 声部割り当てや MIDI 設定の遠隔制御機能: アンサンブル中の他のインストゥルメントから制御できます。
- ・ パネル・スキン: 主にパネル系のモジュールについて、その外観を「スキン」として自由に変えることができるようになりました。フェーダー、ノブ、ボタン、リスト、スイッチ、**Receive**(信号受信)、ランプ、メーターなどで使えます。
- ・ インストゥルメントやマクロの余白: 制御パネルの表示で、周囲に余白を設定できるようになりました。
- ・ オーディオ・ファイルの試聴: ブラウザーやサンプル・マップ・エディターからも試聴できるようになりました。
- ・ 初期設定: イベント入力の初期設定に新しい方式が加わりました。これは REAKTOR 4 との互換モードをオフにした場合に有効です。
- ・ 独自作成ファイルを格納するフォルダー: インストール時にシステム・ファイル (アンサンブル、インストゥルメント、基本マクロ、コア・セル、コア・マクロ) を格納するフォルダーを作成しますが、これとは別に、独自に作成したファイル用のフォルダーも作成、管理するようになりました。アンサンブル、インストゥルメント、基本マクロ、コア・セル、コア・マクロのほか、オーディオ・データ、インポートしたファイル、画像、スナップショット、テーブル・データなども保存しておけます。
- ・ 結線の切断: 接続先の入力ポート上をマウスでつかんでドラッグし、何もない領域で放す、という簡便な方法でも切断できるようになりました。
- ・ デバッグ機能: **Show Event Initialization Order** コマンドで、初期設定の処理順序を確認できるようになりました。
- ・ CPU に対する負荷の表示: 通常は白ですが、CPU に対する負荷のピークが平均以上になった分については黄色、過負荷の場合は赤で表示するようにしました。

1.8. 機能変更

次のように機能を改善しました。

- ・ アンサンブル制御パネル・ウィンドウ：インストゥルメントごとに別々のウィンドウに表示していた内容を、アンサンブルごとに1つの制御パネル・ウィンドウにまとめて見やすくしました。
- ・ ストラクチャー・ウィンドウ：画面が錯綜しないよう、アンサンブル、インストゥルメント、基本マクロ、コア・セル、コア・マクロのどのストラクチャーも、同じウィンドウで編集できるようにしました。但し、**Alt** キーを押しながらアイコンをダブル・クリックするか、**Structure Window** コマンドを実行することにより、別のウィンドウを開いて編集することも可能です。
- ・ ツールバー：ツールバーの構成を見直しました。アンサンブル制御パネル・ウィンドウ、ストラクチャー・ウィンドウにも専用のツールバーを設けたため、主ツールバーから実行する機能は減っています。MacOS X 版の場合、ツールバーは「ツールボックス」の形で、画面上のどこにでも自由に配置できます。これにより、他のウィンドウのヘッダー部分が隠れてしまうこともなくなりました。
- ・ MIDI 信号の入出力状況を表示する、**External MIDI In** および **External MIDI Out** の2つのランプを追加しました。また、コア・ストラクチャーのコンパイル中は、CPU の負荷メーターを流用して、進捗状況を表示します。
- ・ インストゥルメント・ヘッダー：構成を若干変更しました。**A**、**B**、最小化ボタンを左側に移し、ロック / ロック解除の専用ボタン (スパナをかたどったアイコン) を設けました。ミュート / ソロの切り替えボタンは削除しました。また、MIDI 入出力状況を表すランプを、外部 / 内部の MIDI 入力 / 出力で、都合4つにしました。さらに、**In**、**Out** ドロップ・ダウン・メニューに、インストゥルメントの入出力接続 (MIDI、結線) をすべて列挙するようにしました。
- ・ ブラウザー：システム・ファイル、独自に作成したファイルを開くための専用ボタンを設けました。

1.9. 廃止した機能、実行手順を変更した機能

構成変更により必要なくなった、一部の機能を廃止しました。また、実行手順を変更し、使いやすくした機能もあります。

- ・ インストゥルメントごとに別々の制御パネルを設けるのをやめ、アンサンブル制御パネル・ウィンドウにまとめて表示するようにしました。パネルセット・バーを設け、各インストゥルメントに関する情報表示に、クリック1つで切り替えることができるようにしました。

- ・「スクリーンセット」を廃止し、これに代えて、機能を拡充した「パネルセット」を導入しました（「新しい機能」の節も参照）。
- ・これまでブラウザーに組み込まれていた結線機能を、インストールメントのパネル・ヘッダーにある **In**、**Out** メニューに移しました。また、ストラクチャーのブラウズ機能、モジュールの読み込み機能は廃止しました。
- ・インストールメントの内部 MIDI 接続は、プロパティ画面ではなく、**In**、**Out** メニューから実行するようにしました。

1.10. REAKTOR 3 で作成したアンサンブルの取り扱い

REAKTOR 3 で作成したアンサンブルを開くためには、REAKTOR 3 USB コピー保護キーが必要です。これを USB ポートに差し込んだ状態でアンサンブルを開き、REAKTOR 5 の形式で保存し直してください。こうしておけば、次回以降、保護キーは必要ありません。なお、REAKTOR 3 のアンサンブルが多数ある場合は、**Batch Processing** コマンドで一斉に変換するとよいでしょう。

2. 製品の認証手続き

REAKTOR 5 のインストール作業中に**認証手続き**を行い、無期限にこのソフトウェアを使えるようにしなければなりません。この章をざっと読んでからインストール手順を解説した章に進み、その後もう一度この章を読み直すようお勧めします。

2.1. 認証手続きの概要

認証手続きの中でユーザー登録の処理も行います。認証手続きが済めば、この製品に関するオンライン・サービスをすべて利用できるようになります。Native Instrumentsのウェブ・サイトでは、オンライン・チュートリアルを読む、技術サポートを受ける、NI フォーラムに参加する、アップデート用のファイルをダウンロードする、といったサービスを受けることができます。

REAKTOR 5 を今後期限なしに利用するためには、製品の**認証手続き**が必要です。認証しないままでも 30 日間はすべての機能を利用できますが、起動するたびに、認証手続きが済んでいない旨、および利用できる残り日数が表示されます。

認証手続きには **Registration Tool** という小さなアプリケーションを使います。**Registration Tool** は、**認証キー**を受け取るための申し込みコードとして使われる、**System ID** というものを生成します。**System ID** は、コンピューターを構成するハードウェア・コンポーネント、オペレーティング・システムの版番号、REAKTOR 5 のインストール時に入力したシリアル番号をもとに生成されます。

注意：オーディオ・カード、MIDI インターフェイスその他の外部機器を交換しただけならば、新しい**認証キー**を要求されることはありません。コンピューターを構成する基本的なハードウェア・コンポーネントやオペレーティング・システムを入れ替えた場合にのみ、新しい**認証キー**を要求されることになります。この場合、**Registration Tool** を実行するとそれを反映した新しい **System ID** が生成されるので、**認証手続き**をもう一度実行してください。

System ID を Native Instruments に送って**認証キー**を受け取ってください。これで REAKTOR 5 を無期限に使えるようになります。**認証手続き**は利用許諾契約の一部と位置づけられているので、**認証手続き**を

しないまま 30 日が経過すると、REAKTOR 5 を起動することはできなくなります。もちろんこの期間が経過してからでも**認証手続き**をすることは可能です。手続きが完了すればすぐに起動できるようになります。

注意：認証手続きの際、データ通信には 128 ビットのキーを使った暗号化が施されます。Native Instruments では電子メール・アドレスや住所などの個人情報を厳重に管理しており、第三者に開示することはありません。

THIS IS YOUR SYSTEM ID
The System ID is based on certain hardware components of your computer system. During the registration process it will be sent to NI in order to generate an Authorization Key, which will be sent to you per email. (Exchanging your audio card, MIDI interface or external equipment will not require a new Authorization Key.)

1 REGISTRATION
The first step is to send your System ID to Native Instruments and register your product. Choose one of the following options:

A. DIRECT ONLINE REGISTRATION
If you have direct access to the Internet from this computer, this is the fastest way to register your product.

B. SEND REGISTRATION FROM ANOTHER COMPUTER
Use this option if you have access to the Internet via another computer.

C. OFFLINE REGISTRATION
This allows you to fill out an HTML form that can be printed and sent to NI.

2 AUTHORIZATION KEY
Once you receive the email with your authorization key please copy the entire key and paste it into the field below, or open the file that was attached to the mail.

Registration Tool

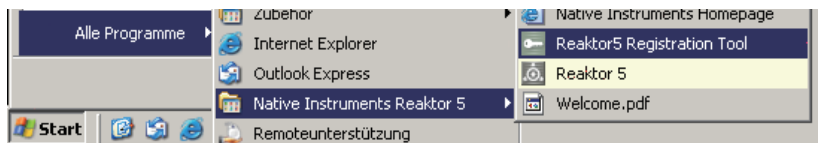
2.2. 認証手続きの手順

Native Instruments では、**認証手続き**を容易にし、不都合を感じさせないようにすることを重視しています。以下、3 通りの手順を解説しますが、中でも**認証キー**を迅速かつ簡便に受け取れる、**A** または **B** の手順をお勧めします。

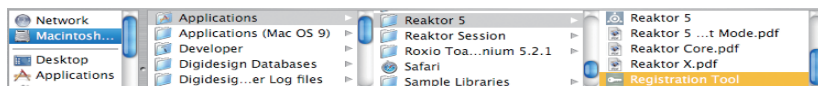
2.3. 手順 A: 稼動コンピューターが直接インターネット接続されている場合

重要：この手順では登録コードを電子メールで送るため、有効な電子メール・アドレスが必要です。

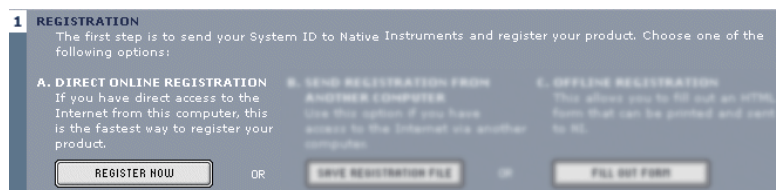
- ・ (Windows の場合): **Registration Tool** を起動します。スタートメニューから **Native Instruments REAKTOR 5 > REAKTOR 5 Registration Tool** を呼び出すか、REAKTOR 5 をインストールしたフォルダー (特に指定しなければ **C:\Program Files\Native Instruments\REAKTOR 5**) から起動してください。



- ・ (MacOS の場合): **Registration Tool** を起動します。REAKTOR 5 をインストールしたフォルダー (特に指定しなければ **Applications\REAKTOR 5**) から起動してください。



- ・ **Register Now** ボタンを押すと、普段使っているウェブ・ブラウザが起動され、Native Instruments の登録ページが開きます。インターネット接続の方法はシステムの設定に従います。さらに、**System ID** が自動的に登録フォームに入力されます。



- ・ 最初のページで、Native Instruments の製品を初めて登録するのか、それとも既に別の製品を登録したことがあり、今回は追加登録するのかを選択します。
- ・ 上記の選択に応じ、新規の場合は住所その他のデータ入力画面、追加登録の場合はユーザー名とパスワードの入力画面が現れます。必要事項を入力し、画面に表示される指示に従って進めてください。

Register Product

Additional Registration: Log in with your existing username and password to add this registration to your account.

First Registration: Your first product registration. Select a new username.

- 登録処理が終わると 12 × 5 桁の**認証キー**がブラウザー画面に表示されます。これをコピーして **Registration Tool** 画面に貼り付けてください。Native Instruments の登録システムから、**認証キー**を通知する電子メールが数分以内に届きます。**認証キー**は電子メールの本文に記載されているほか、同じ**認証キー**を記載したファイルが別に添付されます。さらに、オンライン・サービスに必要なパスワードも書いてあります。

The product is now authorized. The Authorization Key for the current hardware setup is:

07F4D 7090D 5788D 9145D 2830D 8900D
8078D 8257D 0400D 0078D 0078D 0000D

Please copy the entire Authorization Key (both lines) to the clipboard and paste it into the appropriate field of the Registration Tool.

The Authorization Key has also been been sent to the following email address:

mailto:info@nativeinstruments.de

- Registration Tool** をもう一度起動し、**認証キー**を電子メールから転記して **Paste from Clipb.** ボタンを押してください。または、**認証キー**を記載した添付ファイルを別に保存しておき、**Registration Tool** の **Open File** ボタンを押してそのファイルを開く、という方法でも構いません。

2 AUTHORIZATION KEY

Once you receive the email with your authorization key please copy the entire key and paste it into the field below, or open the file that was attached to the mail.

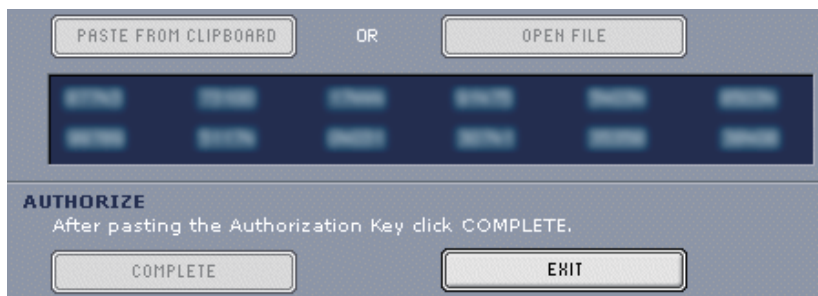
PASTE FROM CLIPBOARD

OR

OPEN FILE

■	■	■	■	■	■
■	■	■	■	■	■

Complete ボタンを押してください。



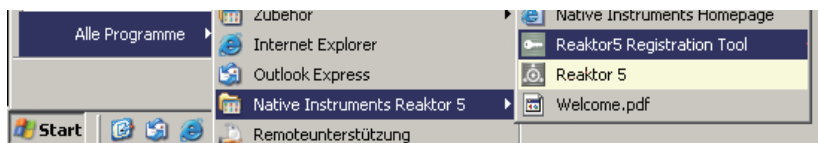
認証手続き終了後の **Registration Tool** の表示

これで REAKTOR 5 の製品**認証手続き**は終了です。認証するよう促すメッセージが起動のたびに現れていましたが、これは出ないようになり、今後は無期限に利用できます。

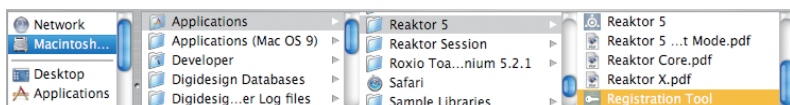
2.4. 手順 B: 別のコンピューターがインターネット接続されている場合

重要: この手順では認証キーを電子メールで送るため、有効な電子メール・アドレスが必要です。

- ・ (Windows の場合): **Registration Tool** を起動します。スタートメニューから **Native Instruments REAKTOR 5 > REAKTOR 5 Registration Tool** を呼び出すか、REAKTOR 5 をインストールしたフォルダー (特に指定しなければ **C:\Program Files\Native Instruments\REAKTOR 5**) から起動してください。



- ・ (MacOS の場合): **Registration Tool** を起動します。REAKTOR 5 をインストールしたフォルダー (特に指定しなければ **Applications\REAKTOR 5**) から起動してください。



- ・ **Save Registration File** ボタンを押すと **Save** ダイアログ・ボックスが現れます。HTML 形式でファイルを保存してください。



- ・ この HTML ファイルを、インターネットにつながっているコンピューターに転送してください。フロッピー・ディスク、CD-R などを使うとよいでしょう。
- ・ 普段使っているブラウザで、この HTML ファイルを開いてください。
- ・ この HTML ページには、Native Instruments のウェブ・サイトにある登録ページへのリンクがあるので、ここをクリックしてください。インターネット接続の方法はシステムの設定に従います。
- ・ 最初のページで、Native Instruments の製品を初めて登録するのか、それとも既に別の製品を登録したことがあり、今回は追加登録するのかを選択します。

Register Product

Additional Registration: Log in with your existing username and password to add this registration to your account.

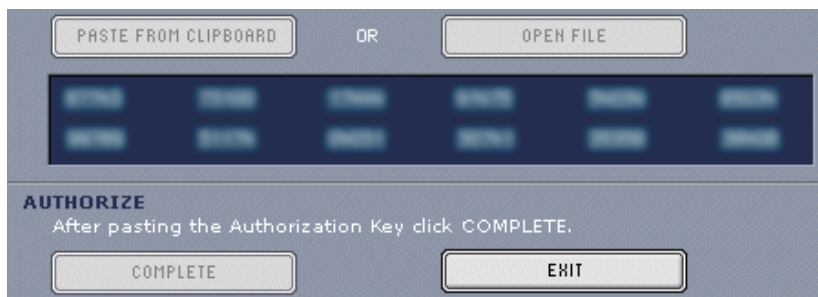
First Registration: Your first product registration. Select a new username.

- ・ 上記の選択に応じ、新規の場合は住所その他のデータ入力画面、追加登録の場合はユーザー名とパスワードの入力画面が現れます。必要事項を入力し、画面に表示される指示に従って進めてください。
- ・ Native Instruments の登録システムから、**認証キー**を通知する電子メールが数分以内に届きます。**認証キー**は電子メールの本文に記載されているほか、同じ**認証キー**を記載したファイルが別に添付されます。さらに、オンライン・サービスに必要なパスワードも書いてあります。

- ・ この添付ファイルを、REAKTOR 5 が動くコンピューターに転送してください。
- ・ **Registration Tool** をもう一度起動し、**Open File** ボタンを押して上記の添付ファイルを開いてください。



- ・ **Complete** ボタンを押してください。



認証手続き終了後の **Registration Tool** の表示

これで REAKTOR 5 の製品**認証手続き**は終了です。認証するよう促すメッセージが起動のたびに現れていましたが、これは出ないようになり、今後は無期限に利用できます。

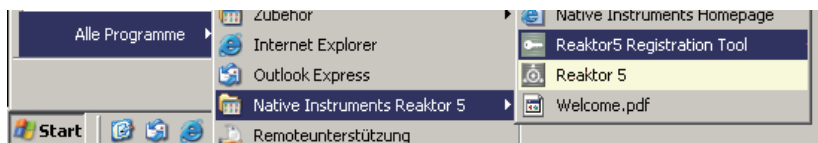
2.5. 手順 C: インターネットにつながっていない場合

重要：この手順では、必要事項を記入した登録用紙を Native Instruments に送る必要があります。**認証キー**の受け取りには、電子メール（推奨）、郵便、ファックスのいずれかを使います。電子メール以外の手段による場合、約 60 桁の**認証キー**を手入力しなければなりません。

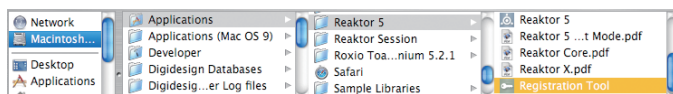
インターネットに接続できない、あるいは有効な電子メール・アドレスを持っていない場合は、郵便またはファックスで**認証手続き**をしてください。Native Instruments ではできるだけ迅速に処理を進めますが、できれば **A** または **B** の手順をお勧めします。その方がすぐに**認証キー**

を受け取れ、操作も簡単だからです。それができない場合は、以下に説明する手順で進めてください。

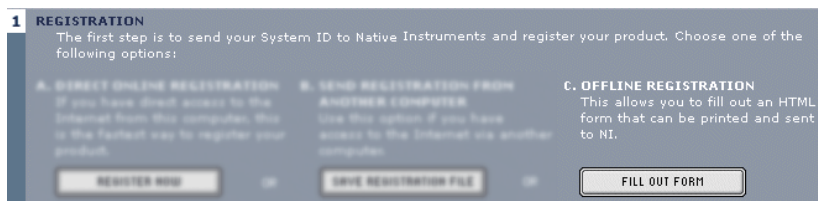
- ・ (Windows の場合): **Registration Tool** を起動します。スタートメニューから **Native Instruments REAKTOR 5 > REAKTOR 5 Registration Tool** を呼び出すか、REAKTOR 5 をインストールしたフォルダー (特に指定しなければ **C:\Program Files\Native Instruments\REAKTOR 5**) から起動してください。



- ・ (MacOS の場合): **Registration Tool** を起動します。REAKTOR 5 をインストールしたフォルダー (特に指定しなければ **Applications\REAKTOR 5**) から起動してください。



- ・ **Fill Out Form** ボタンを押すと HTML ファイルが開きます。オペレーティング・システムの標準ブラウザ、あるいは通常使うと設定されているブラウザが使われます。



- ・ この画面を見ると、認証および登録に必要な事項の入力欄があります。すべて入力してから印刷してください。必要事項がすべて記載されていれば、ほかの書式でも構いません。

手書きの場合は、Native Instruments の登録担当者が読み間違わないよう、読みやすい書体で書いてください。そうしないと認証キーの返送に支障をきたす場合があります。

記入が済んだら、次の宛先に郵送してください。

Native Instruments GmbH (ドイツ)

Registration

Schlesischs Strasse 28

10997 Berlin

Germany

(ファックス: +49 30 6110352400)

Native Instruments USA (アメリカ)

5631 A Hollywood Boulevard

Los Angeles CA 90028

USA

(ファックス: +1-323-372-3676)

- ・ **認証キー**は電子メール(推奨)、ファックス、郵便のいずれかで届きます。
- ・ **Registration Tool** をもう一度起動し、**認証キー**を電子メールから転記して **Paste from Clipb**. ボタンを押してください。または、**認証キー**を記載した添付ファイルを別に保存しておき、**Registration Tool** の **Open File** ボタンを押してそのファイルを開く、という方法でも構いません。**認証キー**を郵便またはファックスで受け取った場合は手入力することになります。
- ・ **Complete** ボタンを押してください。

これで REAKTOR 5 の製品**認証手続き**は終了です。認証するよう促すメッセージが起動のたびに現れていましたが、これは出ないようになり、今後は無期限に利用できます。

2.6. 登録に関する技術サポート

認証手続き中に問題が生じた場合は Native Instruments の登録サポート係にお問い合わせください。Native Instruments のウェブ・サイト (<http://www.native-instruments.com/registersupport.info>) から送れるようになっています。

迅速に解決できるよう、問題の状況その他必要事項をできるだけ正確にお伝えください。

3. Windows XP 版のインストール手順

3.1. 動作条件と推奨システム環境

REAKTOR 5 の動作には、最低でも次の条件を満たすコンピューターが必要です。

ハードウェア

- ・ クロック周波数 1GHz 以上の Pentium III プロセッサーまたは 1.33GHz 以上の Athlon XP プロセッサー。
- ・ 512MB の RAM。
- ・ 1GB 以上のハード・ディスク。
- ・ Windows XP 対応サウンド・カード。
- ・ CD ドライブ。

ソフトウェア

- ・ Windows XP。

重要：REAKTOR 5 の動作には SSE(Streaming SIMD Extensions、マルチメディア拡張命令セット) 対応プロセッサーが必要です。

3.2. ソフトウェアのインストール手順

- ・ REAKTOR 5 CD を CD-ROM ドライブに入れてください。
- ・ Windows エクスプローラーで、CD の内容を表示してください。
- ・ 「REAKTOR 5 Setup.exe」というファイルをダブル・クリックすると、インストール処理が始まります。
- ・ 初期状態では、インストール先フォルダーが C:\Program Files\Native Instruments\REAKTOR 5 となっています。必要であれば変更してください。

インストールされるフォルダー、ファイル、リンク

インストール先フォルダー (Program Files\Native Instruments) の直下に REAKTOR 5 というフォルダーができます。動作に必要なファイルはすべてこのフォルダーの下にあります。**スタート > プログラム > Native Instruments** 以下に、REAKTOR 5 を起動するショートカット

トおよび ReadMe ファイルのメニュー項目が追加されます。但し、インストール処理中にプログラム・グループを変更した場合は、メニュー項目の位置が異なります。

3.3. VST プラグ・インのインストール

- ・ インストール用 CD を CD-ROM ドライブに入れてください。
- ・ Windows エクスプローラーで、CD の内容を表示してください。REAKTOR 5 Setup.exe というファイルをダブル・クリックすると、インストール処理が始まります。
- ・ インストールするコンポーネントの選択画面が現れるので、VST プラグ・インを選択してください。
- ・ VST プラグ・インのインストール先フォルダーを自動で検索するか、あるいは手動で指定するかをここで選択できます。必要に応じて指定してください。

注意: VST 2.0 を使うホスト・プログラムが複数インストールされている場合は、REAKTOR 5 VST.dll というファイルを各ホスト・プログラムの VST プラグ・イン用フォルダーにコピーしてください。Windows で、VST プラグ・イン・ファイルが Windows エクスプローラー上に表示されない場合、**すべてのファイルとフォルダを表示する**よう設定してみてください。この設定は、エクスプローラーの **ツール > フォルダ オプション** コマンドを起動し、ダイアログ・ボックスを **表示** タブに切り替えると、**ファイルとフォルダの表示** という項目の下に見つかります。また、プログラム側の設定により、同じ VST プラグ・インフォルダーを共有するようにすることも可能です。

3.4. DXi 2 プラグ・インの設定

DXi 2 はソフトウェア・シンセサイザーやインストールメント用のプラグ・イン・インターフェイスで、Microsoft DXi 技術にもとづくものです。これに対応したホスト・シーケンサーとしては、Cakewalk 製の Sonar や、Fruity Loops がよく知られています。

インストール手順

- ・ インストール用 CD を CD-ROM ドライブに入れてください。

- ・ Windows エクスプローラーで、CD の内容を表示してください。REAKTOR 5 Setup.exe というファイルをダブル・クリックすると、インストール処理が始まります。
- ・ REAKTOR 5 のインストール処理を続けていくと、**Choose Plugins** という画面が現れます。ここで **Install DXi Plug-in** を選択してください。

インストール・プログラムは REAKTOR 5 プラグ・インをハード・ディスクにコピーし、DXi 2 対応ホスト・プログラムから呼び出して利用できるよう、必要な設定を行います。インストール処理が終わると、REAKTOR 5 がホスト・プログラムのプラグ・インとして表示されるようになります。

3.5. RTAS プラグ・インのインストール

- ・ インストール用 CD にある、REAKTOR 5 Installer を起動します。
- ・ インストール方法として **Custom** を選択してください。
- ・ コンポーネント・リストから、**RTAS** のみを選択してください。

4. MacOS X 版のインストール手順

4.1. 動作条件と推奨システム環境

REAKTOR 5 の動作には、最低でも次の条件を満たすコンピューターが必要です。

ハードウェア

- ・ クロック周波数 1GHz 以上の Apple PowerMac G4。
- ・ 512MB の RAM。
- ・ Core Audio 対応オーディオ・インターフェイス。
- ・ CoreMIDI 準拠の MIDI インターフェイス。MIDI キーボード、MIDI コントローラー、外付けシーケンサーなどの接続用です (スタンドアローン版のみ)。
- ・ 最大 1GB のハード・ディスク。
- ・ CD ドライブ。

ソフトウェア

- ・ MacOS 10.2.6。

重要 : REAKTOR 5 の動作には AltiVec(ベクトル演算ユニット) 搭載プロセッサが必要です。

4.2. ソフトウェアのインストール手順

- ・ インストール用 CD を CD-ROM ドライブに入れてください。
- ・ Install REAKTOR 5 という名前のプログラムをダブル・クリックすると、インストール処理が始まります。
- ・ まず開始画面が現れます。**Continue** ボタンを押すと利用許諾契約が表示されるので確認してください。次に、インストール先フォルダーを選択する画面になります。

インストール先フォルダーの候補が提示されます。特に変更しなければ、最初に見つかったハード・ディスク上になります。また、**Easy Install** を指定すればスタンドアローン版とプラグ・イン版の両方がインストールされ、**Custom Install** ならばいずれか一方の版を選択できます。

4.3. MacOS Audio Unit プラグ・インのインストール

- ・ インストール用 CD にある、REAKTOR 5 Installer を起動します。
- ・ インストール方法として **Custom** を選択してください。
- ・ コンポーネント・リストから、**Audio Unit** のみを選択してください。

4.4. RTAS プラグ・インのインストール

- ・ インストール用 CD にある、REAKTOR 5 Installer を起動します。
- ・ インストール方法として **Custom** を選択してください。
- ・ コンポーネント・リストから、**RTAS** のみを選択してください。

5. オーディオ・インターフェイス

オーディオ・インターフェイスには「ドライバー」というソフトウェアが含まれています。REAKTOR 5 を始めとするさまざまなプログラムは、これを介してオーディオ・ハードウェアと情報をやり取りします。ここでは各種のオーディオ・インターフェイスを REAKTOR 5 から使う手順を解説します。

REAKTOR 5 の使用形態には、大きく分けて「スタンドアローン」と「プラグ・イン」の2通りがあります。

「スタンドアローン」の場合、その名前の通り、ホスト・ソフトウェアなしで単独で動作します。REAKTOR 5 のオーディオ /MIDI 入出力は、コンピューターに装着されたオーディオ /MIDI ハードウェア・インターフェイスと直接やり取りします。

一方「プラグ・イン」の場合は、シーケンサー、ハード・ディスク・レコーダーなどの「ホスト」プログラムに組み込まれる形で動作します。コンピューターに装着されたオーディオ /MIDI ハードウェアとは、このホスト・プログラムが直接やり取りします。REAKTOR 5 とホスト・プログラムは、「仮想パッチ・コード」を使って接続されます。REAKTOR 5 のオーディオ出力は、ミキサー信号の形でホスト・プログラムに入ります。逆にホスト・プログラムは MIDI データを REAKTOR 5 に送ります。

それぞれの使用形態について詳しく解説する前に、オペレーティング・システムやホスト・プログラムごとに、どのようなドライバーやプラグ・インが使えるかを紹介します。

スタンドアローンでの使い方

スタンドアローンの場合、ASIO、MME、DirectSound、Core Audio などのドライバーが使えます。REAKTOR 5 が動作するコンピューターは、ハードウェア・ディジタル・シンセサイザーと同格の、独立した楽器（インストゥルメント）として動作します。オペレーティング・システムごとに、どのドライバーが使えるかを表に示します。

インターフェイス / ドライバー	Windows	MacOS X
ASIO 2.0	○	○ (10.2 以降)
DirectSound	○	-
MME	○	-
Core Audio	-	○

プラグ・インとしての使い方

REAKTOR 5 は独立して稼動するのではなく、シーケンサーなどの「ホスト」プログラムに組み込まれる「モジュール」として動作します。ホスト・プログラムと緊密に連携して動作するのに加え、次のような特徴があります。

- ・ REAKTOR 5 側で MIDI の曲作りをし、次に MIDI トラックをミックス・ダウンする処理が、ひとつのプログラム内で完結します。
- ・ シーケンサー側のオートメーション機能で REAKTOR 5 のパラメーターを制御できます。
- ・ ほかにプラグ・インを組み合わせ、REAKTOR 5 の出力信号をさらに加工することができます。
- ・ MIDI コントローラーとの間で、サンプル単位の正確なタイミング調整が可能です (VST 2.0 プラグ・インとして使う場合)。
- ・ ホスト・プログラム側のファイル (シーケンサーの「ソング」ファイルなど) に、プラグ・イン側のパラメーター設定なども一括して保存しておけます。
- ・ 他のインストゥルメントとも組み合わせて「仮想スタジオ」を構築できます。

ショートカットやキー・コマンドは機能しません。ホスト・プログラムがキー入力を取ってしまうと、プラグ・インには渡さないからです。

各種のインターフェイスがどのようなホスト・プログラムに対応しているか、その概要を表に示します。

インターフェイス	ホスト・プログラム	Windows	Mac
VST 2.0 プラグ・イン	C u b a s e 、 Nuendo	○	○
DXi	Sonar	○	-
Audio Units	Logic	-	○

注意：ホスト・プログラムによっては、プロトコルを実行時に選択できる「ラッパー」を用意している場合があります。それぞれを試し、性能的に優っている方を使ってください。例えば Sonar の場合、DXi プロトコルならば複数の出力が使える、一方 VST であればオートメーション関係の機能が充実しています。

インターフェイスの詳細

以下に紹介するのは、REAKTOR 5がサウンド・カードと情報をやり取りするインターフェイスの種類です。コンピュータの機種やオーディオ・インターフェイス (サウンド・カード)、オペレーティング・システム (Windows XP または MacOS X) によって、実際に利用できるインターフェイスは異なります。使えるものの中からできるだけ処理性能に優れたものを選ぶこととなりますが、多くの場合、Windows ならば ASIO、Mac ならば Core Audio が適しているでしょう。Windows では DirectSound や Multimedia (別名 MME) も使えますが、演奏してから実際に音が出るまでの遅延が大きい傾向があるのでお勧めできません。

- ・ **ASIO** (Audio Streaming Input Output): Windows にも MacOS にも対応したプロトコルで、Steinberg が開発しました。遅延が小さく、マルチ・チャンネルのオーディオ・カードにも対応しており、処理性能も優れているのでお勧めです。
- ・ **DirectSound**: Microsoft が開発した Windows 用インターフェイスで、DirectX 5.0 以降のコンポーネントです。DirectX が正常に使えるかどうかは、サウンド・カードに依存します。遅延が小さくなるよう無理に調整すると、障害や雑音が発生する恐れがあります。
- ・ **MME** (Multi Media Extension): Windows 用の標準的なオーディオ・ドライバです。ほとんどのサウンド・カードが対応しており、かなり良好に動作します。但し遅延が大きいので、リアル・タイム・アプリケーションで使う場合は DirectSound よりも劣ります。
- ・ **Core Audio**: MacOS X 用のオーディオ・インターフェイスで、オペレーティング・システムに緊密に統合されています。Mac 内蔵オーディオ出力ばかりでなく、外付けオーディオ・ハードウェアにも対応しています。

プラグ・インの詳細

- ・ **VST** (Virtual Studio Technology): ASIO と同じく、Windows にも MacOS にも対応したプラグ・イン技術で、Steinberg が開発しました。最も普及しているフォーマットなので、多くのプログラムが VST プラグ・イン用に最適化されています。

- ・ **DXi2** (DirectX Instrument 2): Microsoft の DirectX 技術にもとづく Windows 用のプラグ・イン・インターフェイスで、遅延を小さくし、処理性能を改善することを目標として設計されました。DXi に対応したホスト・シーケンサーとしては、Cakewalk の Sonar や Image Line の Fruity Loops Studio がよく知られています。
- ・ **RTAS** (Real Time Audio Suite): DigiDesign が提唱しているインターフェイス・プロトコルで、ProTools 用プラグ・イン (や DigiDesign と互換性のある他のソフトウェア) を利用できます。従来、TDM エフェクトは Digidesign 製のハードウェアと組み合わせて使うことしかできませんでしたが、RTAS プラグ・インは特定のハードウェアに依存することなく、必要な計算処理をすべてホスト・プロセッサ側で実行するようになっています。
- ・ **AU** (Audio Units): MacOS X 専用のフォーマットで、オペレーティング・システムに緊密に統合されています。

遅延についての追記

オーディオ信号をデジタル・データに、あるいはその逆に変換する機器は一般に、その処理の過程で多少の遅延が生じます。ハードウェアで処理を行う DSP (デジタル・シグナル・プロセッサ) も同様です。現在では、コンピュータの処理能力向上と遅延の小さなサウンド・カード・ドライバの登場により、ちょっと聴いただけでは気づかない程度の遅延になっています。具体的には 3 ミリ秒以下で、これはスピーカーから耳までの距離を 1 メートルほど遠くした場合に加わる程度の遅延に過ぎません。しかし一般のコンピュータは、遅延を小さく抑えるような設定にはなっていないことが多いので、生演奏などの場合にこれが気になることもあります。

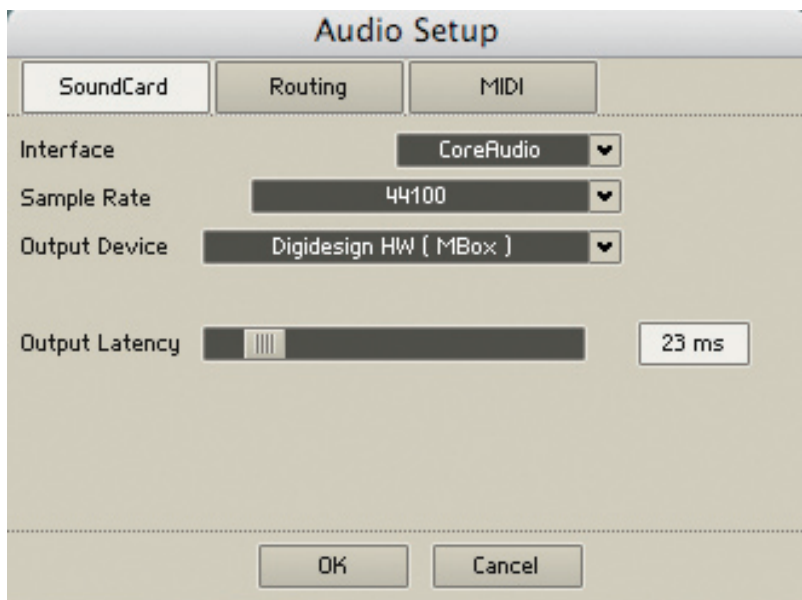
6. スタンドアローン版の REAKTOR 5

REAKTOR 5 をプラグ・インとして使う場合、オーディオ /MIDI の接続はホスト・プログラムが済ませてしまっており、そこに「差し込まれる」形で動作します。これに対してスタンドアローンの場合は、REAKTOR 5 が直接オーディオ・インターフェイスと通信します。したがって、オーディオ /MIDI 設定やドライバーとの通信プロトコル設定が必要になります。当然、プラグ・インの形式を設定する必要はありません。

特に明示している点を除き、Mac 版でも Windows 版でも使い方は同じです。オーディオ・インターフェイスを変えると、多くの場合こういった設定も変える必要があるので注意してください。

設定は **Audio Setup** 画面で行います。**Setup** メニューから開いてください。この画面は **Soundcard**、**Routing**、**MIDI** の 3 つのタブに分かれています。オーディオ出力のつなぎ替えには **Routing** タブを使います。

6.1. Soundcard タブ (オーディオ・インターフェイス)



- ・ **Interface:** オーディオ・インターフェイス用プロトコルのうち、できるだけ高速に動作するものを選んでください。通常、ASIO か Core Audio のいずれかになるでしょう。Windows では DirectSound や Multimedia (別名 MME) も選択肢にありますが、演奏してから実際に音が出るまでに、かなりの遅延が生じます。

(Windows のみ) 「emulated」と付記されているドライバーは性能が劣るので使わないでください。例えば DirectSound は一般に MME より高性能なのですが、エミュレートされた DirectSound ドライバーは逆に MME よりも劣ります。

- ・ **Sample rate:** オーディオ・インターフェイスが対応可能なサンプル・レートが、ドロップダウン・メニューに列挙されています。44.1kHz は CD にも使われているもので、最も一般的な選択肢といえます。しかし 48kHz や 96kHz のサンプル・レートに対応したオーディオ・インターフェイスもあります (REAKTOR 5 は 96kHz にも対応しています)。このような高いサンプル・レートにすると CPU に対する負荷は増しますが、高い周波数の信号に対する応答性を若干改善することができます。REAKTOR 5 をスタンドアローンで使う場合は、必要に応じてどのサンプル・レートにしても構いません。しかし (Cubase、Digital Performer、Logic、Sonar などの) プラグ・インとして使う場合は、ホスト・プログラム側にも依存してサンプル・レートが決まります。
- ・ **Output Device:** 使用するオーディオ・インターフェイス用の ASIO を指定します (「ASIO DirectX」や「ASIO Multimedia」ではありません。但しほかに選択肢がない場合はこれを指定しても構いません)。Mac の場合は Core Audio になります。
- ・ **Output Latency:** 出力の遅延が表示されます。一部のドライバーでは、フェーダーを使って個別に調整できます。
遅延時間は、安定したオーディオ性能が得られる範囲内で、できるだけ短い値に設定してください。この値が小さ過ぎて CPU の処理が間に合なくなると、「パチン」とか「ポン」とかいう雑音が混じることがあります。逆に大きな値を設定するとオーディオ性能は安定しますが、音楽的に不自然になってしまう恐れがあります。

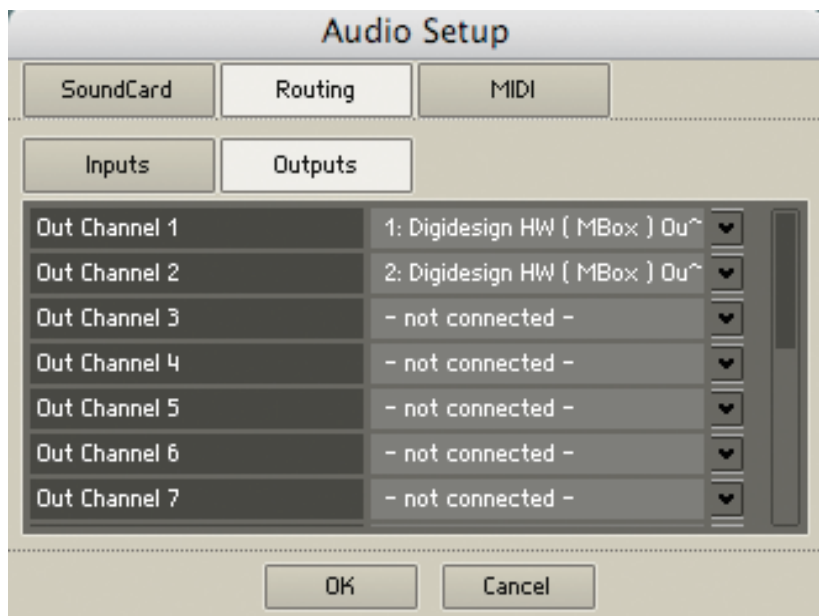
いろいろな設定を試して、オーディオ性能の安定性と応答性のバランスが最もよい値を見つけてください。次のような手順で調整するのが手軽でしょう。

適当なインストゥルメントを選び、再生しながら **Output Latency** スライダーを動かします。

左方向にゆっくり動かし、オーディオ出力にクリック雑音が混じり始めたところでいったん止めてください。

次に右方向にゆっくり動かし、クリック雑音が消えたところで止めてください。これが最適な値です。

6.2. Routing タブ



ドロップ・ダウン・メニューを使って、REAKTOR 5 の出力と、マルチ出力サウンド・カードの出力とを対応づけることができます。

入出力を複数備えたサウンド・カードの場合、どれと REAKTOR 5 を接続するか、この画面で設定できます。**Inputs** ボタン、**Outputs** ボタンを押すと、それぞれドロップ・ダウン・メニューが開くので、接続先を選択してください。なお、左右のチャンネルは独立なので、ステレオ・ペア以外の割り当ても可能ですし、片方だけ切り離しても構いません。

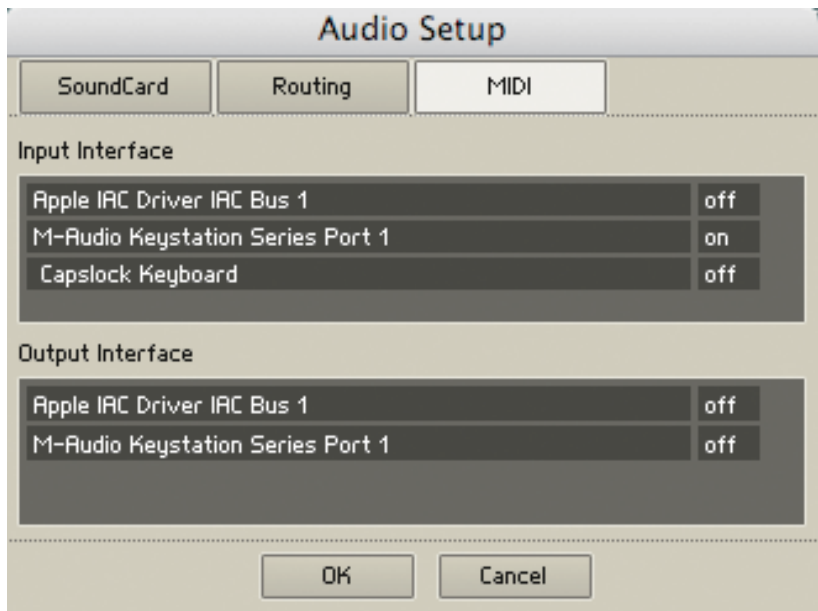
REAKTOR 5 には単声の出力端子が 8 つあります。すべてをスタンドアローン・モードで使うためには、出力チャンネルを 8 つ備えたオーディオ・インターフェイスが必要です。

もっとも、これだけのチャンネルが常に必要というわけではありません。例えばノート型パソコンで、組み込みオーディオ・カードにステレオ出力が 1 系統 (2 チャンネル) しかない場合、REAKTOR 5 の入出力すべてをこのステレオ出力に割り当てることも可能です。

各ドロップ・ダウン・リストには、**SoundCard** タブの **Interface** で選択したドライバー / オーディオ・インターフェイスで使える入出力がすべて列挙されます。REAKTOR 5 のチャンネル 1/2(右)、チャンネル 3/4(左)、チャンネル 3/4(右)、チャンネル 5、チャンネル 6、チャンネル 7、チャンネル 8 のそれぞれに、ハードウェアの出力を割り当ててください。

(Windows のみ) オーディオ・インターフェイス全体の音量レベルは、ハードウェア付属のミキサー・アプレットや、Windows の組み込みミキサーで調整できます。音量が異常に大きい / 小さい場合は、**スタートメニューからプログラム > アクセサリ > エンターテインメント > ボリューム コントロール**を起動し、WAVE の音量スライダーを調整してみてください。

6.3. MIDI タブ



MIDI タブに切り替えると MIDI I/O が列挙されており、初期状態ではいずれも **off** になっています。この欄をクリックするごとに、**on** と **off** が交互に切り替わります。

複数の入力を **on** にした場合、各入力信号はマージされます。


7. プラグ・イン版の REAKTOR 5


プラグ・インとして REAKTOR を使う場合、スタンドアローン版とは多少見え方が異なりますが、プラグ・インとしては無意味なものを除き、すべての機能を利用できます。

コマンドは一般に、コンテキスト・メニューから起動するようになっています。ツールバーの背景部分 (ボタン類がない部分) で、Windows ならば右クリック、MacOS X ならば **control** キーを押しながらクリックすると、コンテキスト・メニューが現れます。ここにはツールバーを隠す (非表示にする) コマンドもあります。ツールバーが現れていなくても、ブラウザーやスナップショット、プロパティー画面の背景部分で同様に操作すれば、コンテキスト・メニューを呼び出すことができます。

ツールバーを非表示にし、アンサンプル・ウィンドウ以外をすべて閉じた状態で **Resize** ボタンを押すと、アンサンプル全体がプラグ・イン・ウィンドウにちょうど納まるようになります。

プラグ・イン・ウィンドウの左側は、ビューを切り替えながら表示するようになっています。すなわち、**F4** キーでプロパティー、**F5** キーでブラウザー、**F6** キーでスナップショットの各表示に切り替わります。また、小さな「×」がついた **Close** ボタンを押すと、この左側領域が非表示になります。

ツールバーが表示されている場合、その右側にある **Browser** ボタン  を押すとブラウザー表示になります。ツールバーが出ていない場合は **F5** キーで切り替えてください。

スナップショットの表示に切り替えるには、**Snapshot** ボタン  を押してください。これはアンサンプル・ウィンドウやインストゥルメント・ヘッダーにあります。**F6** キーでも切り替えることができます。

プロパティー表示には、インストゥルメント制御パネル内のつまみをどれでもダブル・クリックすれば切り替えられます。**F4** キーでも切り替えることができます。

サンプル・マップ・エディターはプラグ・イン・ウィンドウの下側に表示されます。このエディター画面を開くには、インストゥルメント制御パネルにあるサンプラーの波形表示部分をダブル・クリックしてください。

アンサンブルの保存には、ツールバー上の **Save** ボタンを使います。**Ctrl** キーを押しながら **Save** ボタンを押すと、ファイル名を変えて保存できます。

オートメーション：ホスト・アプリケーションがプラグ・インのオートメーションに対応している場合、REAKTOR はホスト・アプリケーション側に、アンサンブルに使われている各種パラメーターの名前と値の範囲を通知します。

7.1. オートメーション ID の編集

アンサンブルに関して制御可能な項目には、一意的なオートメーション ID が割り当てられており、これを使ってホスト・アプリケーション側から制御できるようになっています。また、ホスト側のパラメーター・リストには、この ID の順に項目が並びます。そこで、ID を割り当て直すための機能をいくつか設けました。扱えるパラメーターの個数に制限があるホスト・アプリケーションの場合は特にこの機能が重要です。

- **Compress:** ID 番号に飛びがあればこれを詰めます。
- **Sort:** 同じマクロに関する制御項目がひとかたまりになるよう、ID を割り当て直すことができます。したがってパラメーター・リストの表示順序もマクロごとになります。
- **Instrument up, Instrument down:** アンサンブルに属する各インストゥルメントには、一意的なベース ID が割り当てられます。パラメーター・リストの制御項目は、このベース ID によって大きくグループ分けして表示されます。その順序を変更するための機能です。

7.2. ファイル管理、特に自動保存の機能について

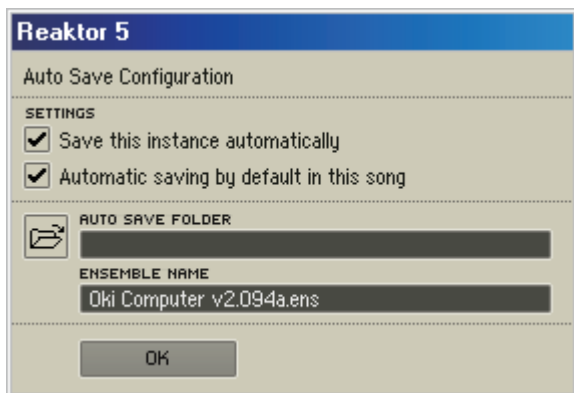
REAKTOR プラグ・インの読み込み

ホスト・アプリケーションに REAKTOR プラグ・インを挿入すると、**New Ensemble**、**Load Ensemble...** の 2 つの選択肢が現れます。

New Ensemble を選択すると、New.ens という空のストラクチャーが読み込まれた状態になり、ここに独自のアンサンブルを構築していくことになります。**Load Ensemble...** は作成済みのアンサンブルをファイルから読み込む場合に選択します。

ブラウザーその他から、ドラッグ & ドロップ操作でアンサンブルを読み込むことも可能です。

自動保存機能の設定

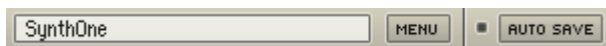


アンサンプルを読み込むか新たに作成すると、**Auto Save Configuration** ダイアログ・ボックスが開き、次のような設定ができます。

- ・ **Save this instance automatically:** 特に必要がなければ、オンのままにしておくようお勧めします。このアンサンプルをあるプロジェクトに取り込み、プロジェクトを保存しようとする、自動的に元のアンサンプルを、独立したファイルに複製保存するようになります。あとで元のアンサンプルを修正しても影響はないので安全です。但し、保存先フォルダーとファイル名を設定しておく必要があります。一方、オフにすると複製保存されなくなるので、ディスク容量の節約になります。プロジェクト・ファイルに保存されるのは、アンサンプルの制御パネルで設定した内容だけです。こうすると複数のプロジェクトで同じアンサンプルを共有できる代わりに、あるプロジェクト上でアンサンプルを修正すると、他のプロジェクトにも影響が及びます。さらに、プロジェクトを保存しても、イベント / オーディオ・テーブル・データや、サンプラー・ファイルの参照情報が、保存の対象にならない点にも注意が必要です。
- ・ **Automatic saving by default in this song:** これをオンにすると、プロジェクトに REAKTOR プラグ・インを追加すること、**Auto Save Configuration** ダイアログ・ボックスが開くようになります。プロジェクトに属するアンサンプルの状態を、確実にそれぞれのファイルに保存できます。

- ・ **AUTO SAVE FOLDER:** このアンサンブルの複製を保存するフォルダーの設定です。キーボードで直接入力するほか、フォルダー型のアイコンがついたボタンを押し、ファイル・ブラウザを開いて指定しても構いません。プロジェクト・ファイルと同じフォルダーを指定しておけば、プロジェクトごとまとめてコピーするような場合に便利です。
- ・ **ENSEMBLE NAME:** 複製を保存するファイル名の設定です。他のファイルと混同することのないよう、一意的な名前を指定してください。
- ・ **OK:** これを押すと上記で設定した事項が確定します。なお、自動保存の機能がオンになっているのにフォルダー名やファイル名が未設定であれば、このボタンは押せません。
- ・ **Cancel:** 設定した事項を取り消し、このダイアログ・ボックスを閉じます。

プラグ・イン・ヘッダーから呼び出せる自動保存機能



- ・ 一番左の欄は、現在開いているアンサンブルのファイル名を表します。**Auto Save Configuration** ダイアログ・ボックスでファイル名を変更すれば、それが反映されます。
- ・ **MENU** ボタンを押すと主メニューが開きます。
- ・ これと **AUTO SAVE** ボタンの間にあるランプは、プラグ・インが自動保存の対象であるかどうかを表します。
- ・ **AUTO SAVE** ボタンを押すと **Auto Save Configuration** ダイアログ・ボックスが開き、自動保存に関する設定を変更できます。保存先フォルダーを変更すると、該当するファイルが新しいフォルダーに移動されます。このとき、元のフォルダーにほかにもファイルがあれば、その保存先フォルダーも変更するかどうか訊ねられます。

アンサンブルの置き換え

自動保存がオンであれば、アンサンブルを置き換えようとするとき **Auto Save Configuration** ダイアログ・ボックスが開きます。したがって、いくつものアンサンブルを切り替えて比較しようという場合は、自動保存をオフにしておいた方がよいでしょう。

読み込もうとしたアンサンプルが見つからない場合

プロジェクトを開く際、これに属するアンサンプル・ファイルが見つからなければ、プラグ・イン・ウィンドウにその旨の表示が現れます。**Locate Ensemble** ボタンでファイル・ブラウザーを開く、(他の場所にある) ファイルをウィンドウ上にドラッグする、**Auto Save Configuration** ダイアログ・ボックスでフォルダー / ファイルを再設定する、といった方法で対処してください。フォルダーを変更した場合、ほかにもファイルが見つからないアンサンプルがあれば、新しいフォルダーを検索し直すこともできます。

複数のプラグ・インの切り替え

別の REAKTOR プラグ・インをプロジェクトに読み込むと、自動保存の可否や保存先フォルダーは、新しいプラグ・インの設定に置き換わります。

プロジェクトに保存される内容

プロジェクトを保存しようとした (あるいはホスト側からプラグ・イン・データを要求された) 場合の処理は、自動保存のオン / オフにより次のように異なります。

- ・ オンであれば、アンサンプルを、所定のフォルダーに、ファイルの形で保存します。
 - ・ オフであれば、アンサンプル・ファイルの場所 (パス) を、プロジェクト・データとして保存します。
- いずれの場合も、次のデータを併せて保存します。
- ・ 制御パネルで設定した事項。
 - ・ 自動保存の可否、保存先フォルダーの設定。
 - ・ ウィンドウの大きさと状態 (最小化、自動配置、固定配置)。

アンサンプルの複製の保存

プラグ・イン用の **Save As** コマンドは **Save A Copy As** コマンドに変わりました。アンサンプルの複製を保存しますが、自動保存用のフォルダーやファイル名には影響しません。

自動保存のオン / オフのデフォルト設定

環境設定画面で **Auto Save off by default** をオンにすると、アンサンプルを新たに作ったとき、自動保存がオフの状態になります。

プラグ・イン・ウィンドウの大きさ制御機能



ウィンドウの大きさを制御するための機能がいくつかあります。

これはプラグ・イン・ツールバーの右端にある 4 つのボタンで制御します。左側 2 つは、最小化および最大化ボタンです。3 つ目のボタンを押すと、ウィンドウの中身がちょうど納まる大きさになります。4 つ目は自動大きさ調整のオン / オフ切り替えで、オンにしておくと、中身の切り替え、A/B パネルの切り替えなど、必要に応じて自動的に大きさが変わるようになります。

なお、最大化したときの実際の大きさは、環境設定として指定できるようになりました。

7.3. VST 2.0 プラグ・イン

7.3.1. Cubase SX 3 でのプラグ・インの使い方

- ・ Cubase を起動し、**Devices** メニューから **VST Instruments** コマンドを実行してください。F11 キーでも同じ操作ができます。
- ・ インストゥルメントの「収納庫」をかたどったウィンドウが現れます。空きスロットをクリックするとプラグ・インの一覧が現れるので、REAKTOR 5 を選択してください。



- ・ 選択したプラグ・インがリストに現れ、自動的にオンになっています。また、VST ミキサーには、ミックス・ダウンに必要なオーディオ・チャンネルが生成されます。これにより、Cubase ソングにある他のオーディオ・トラックと同じように、REAKTOR 5 の出力をミックスし、パンを調整し、さまざまな加工を施すことができます。

- ・ **Edit** ボタンを押すと REAKTOR 5 の操作画面が現れ、必要な調整や編集ができるようになります。
- ・ **Project** ページを開き、MIDI トラックがなければ追加してください。



- ・ この MIDI トラックの **Output** パラメーター編集領域を開き、該当する欄をクリックすると、割り当て可能な MIDI 出力ポートのリストが表示されます。**REAKTOR 5** を指定してください。

注意：このリストに REAKTOR 5 が現れない場合、正しくインストールされていない恐れがあります。インストール手順についての章 (Windows 用または Mac 用) を参照し、もう一度確認してください。

ライブラリーからインストールを読み込むと、キーボード・コントローラー側から MIDI 信号を送れば演奏できる状態になります。REAKTOR 5 が生成する音声信号は、VST ミキサーを経由して直接サウンド・カードに送られます。MIDI 信号を受け取らない、あるいは音声信号を生成しない場合は、次の点を確認してみてください。

- ・ Cubase 側で「MIDI thru」が有効になっていること。
- ・ MIDI トラックの MIDI チャンネルが、インストールメントの受信チャンネルと一致していること。
- ・ サウンド・カードを Cubase 用に正しく設定していること。
(詳しくは Cubase のマニュアルも参照してください)

7.3.2. Nuendo 2.0 でのプラグ・インの使い方

- ・ Nuendo を起動し、空のプロジェクトを作るか、既存のプロジェクトを立ち上げてください。
- ・ **Devices** メニューから **VST instruments** コマンドを実行してください。**F11** キーでも同じ操作ができます。
- ・ インストールメントの「収納庫」をかたどったウィンドウが現れます。空きスロットをクリックするとプラグ・インの一覧が現れるので、REAKTOR 5 を選択してください。



- ・ 選択したプラグ・インがリストに現れ、自動的にオンになっています。また、VST ミキサーには、ミックス・ダウンに必要なオーディオ・チャンネルが生成されます。これにより、Nuendo プロジェクトにある他のオーディオ・トラックと同じように、REAKTOR 5 の出力をミックスし、パンを調整し、さまざまな加工を施すことができます。
- ・ **Edit** ボタンを押すと REAKTOR 5 の操作画面が現れ、必要な調整や編集ができるようになります。
- ・ **Project Editor** ページを開き、MIDI トラックがなければ追加してください。
- ・ この MIDI トラックの **Output** パラメーター編集領域を開き、該当する欄をクリックすると、割り当て可能な MIDI 出力ポートのリストが表示されます。**REAKTOR 5 VST** を指定してください。なお、MIDI 入力ポートは、MIDI コントローラー側の設定に合わせて割り当てる必要があります。



- ・ これで MIDI トラックに録音する準備ができました。

注意：このリストに REAKTOR 5 が現れない場合、正しくインストールされていない恐れがあります。インストール手順についての章 (Windows 用または Mac 用) を参照し、もう一度確認してください。

ライブラリーからインストールメントを読み込むと、キーボード・コントローラー側から MIDI 信号を送れば演奏できる状態になります。REAKTOR 5 が生成する音声信号は、VST ミキサーを経由して直接サウンド・カードに送られます。MIDI 信号を受け取らない、あるいは音声信号を生成しない場合は、次の点を確認してみてください。

- ・ Nuendo 側で「MIDI thru」が有効になっていること。
- ・ MIDI トラックの MIDI チャンネルが、インストールメントの受信チャンネルと一致していること。
- ・ サウンド・カードを Nuendo 用に正しく設定していること。
(詳しくは Nuendo のマニュアルも参照してください)

7.4. Audio Units プラグ・イン

7.4.1. Logic 7.x でのプラグ・インの使い方

- ・ Logic を起動し、オーディオ・インストゥルメント・トラックを生成するか、または既存のオーディオ・トラックか MIDI トラックを割り当ててください。割り当てたいトラックをクリックするとメニューが現れるので、マウス・ボタンを押したまま動かして、**Audio > Audio Instrument > Inst 1** を選択します。



- ・ オーディオ・インストゥルメント・トラックをダブル・クリックすると Environment ウィンドウが現れます。自動的にスクロールされ、Logic ミキサーのインストゥルメント・バスのうち先頭のものが表示されます。
- ・ REAKTOR 5 Audio Unit プラグ・インを、インストゥルメント・ミキサー・バスの適当なインサート・スロットに割り当てます。割り当てたいスロット上をクリックするとメニューが現れるので、**Stereo > Audio Units > Native Instruments > REAKTOR 5** を選択してください (REAKTOR 5 はマルチ・チャンネル・インサートとしても使用できます)。



- ・ 以上の操作により、プラグ・インがインストゥルメント・スロットに表示され、待機状態になっています。これにより、Logic の他のオーディオ・トラックと同じように、REAKTOR 5 の出力をミックスし、パンを調整し、さまざまな加工を施すことができます。
- ・ REAKTOR 5 の操作画面が開いていない場合は、ミキサーの「REAKTOR 5」スロットをダブル・クリックしてください。これで必要な調整や編集ができるようになります。

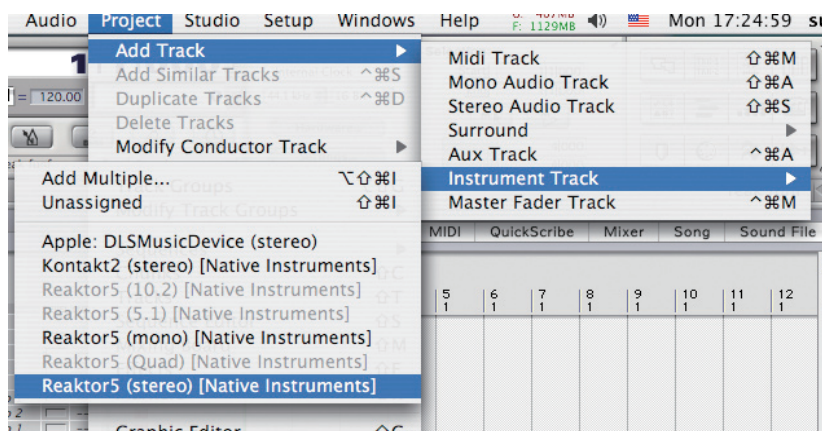
注意：Audio Units ホスト・アプリケーション側の AU インストゥルメント・リストに REAKTOR 5 が現れない場合、正しくインストールされていない恐れがあります。インストール手順についての章 (Mac 用) を参照し、もう一度確認してください。

ライブラリーからインストゥルメントを読み込むと、キーボード・コントローラー側から MIDI 信号を送れば演奏できる状態になります。REAKTOR 5 が生成する音声信号は、VST ミキサーを経由して直接サウンド・カードに送られます。MIDI 信号を受け取らない、あるいは音声信号を生成しない場合は、次の点を確認してみてください。

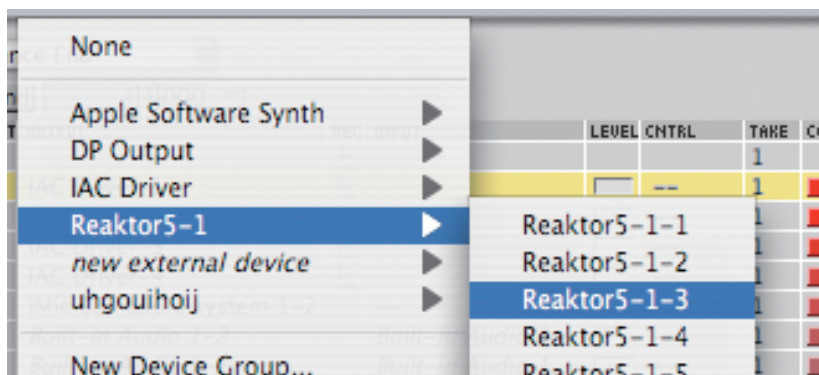
- ・ アレンジ・ウィンドウでインストゥルメント・トラックが選択されていること。
- ・ MIDI トラックの MIDI チャンネルが、インストゥルメントの受信チャンネルと一致していること。
- ・ サウンド・カードを Logic 用に正しく設定していること。
(詳しくは Logic のマニュアルも参照してください)

7.4.2. Digital Performer 4.5 以降でのプラグ・インの使い方

- ・ Digital Performer を起動し、**Project > Add Track > Instrument Track > REAKTOR 5** コマンドでインストゥルメント・トラックを作成してください。



- ・ **Project > Add Track > Midi Track** コマンドで MIDI トラックを作成してください。Digital Performer のトラック一覧ウィンドウ (またはシーケンス・エディター・ウィンドウ) で、この MIDI トラックの出力を「REAKTOR 5-1」の MIDI チャンネルに割り当てておきます。なお、ほかにも REAKTOR 5 プラグ・インを使う場合、名前は「REAKTOR 5-2」、「REAKTOR 5-3」などとなります。



- ・ これでプラグ・インが使えるようになりました。Digital Performer のミキサーを使って、他のオーディオ・トラックと同じように REAKTOR 5 の出力をミックスし、パンを調整し、さまざまな加工を施すことができます。
- ・ キーボードで REAKTOR 5 を演奏したい場合は、REAKTOR 5 に接続された MIDI トラックを録音可にしてください。なお、Digital Performer の **Studio** メニューで、**Midi Patch Through** をオンにしておく必要があります。
- ・ Digital Performer のミキシング・ボード上で REAKTOR 5 スロットをダブル・クリックすると、REAKTOR 5 の制御画面が開き、ここですべての機能を制御、編集できます。

注意：Audio Units ホスト・アプリケーション側の AU プラグ・イン・リストに REAKTOR 5 が現れない場合、正しくインストールされていない恐れがあります。インストール手順についての章 (Mac 用) を参照し、もう一度確認してください。

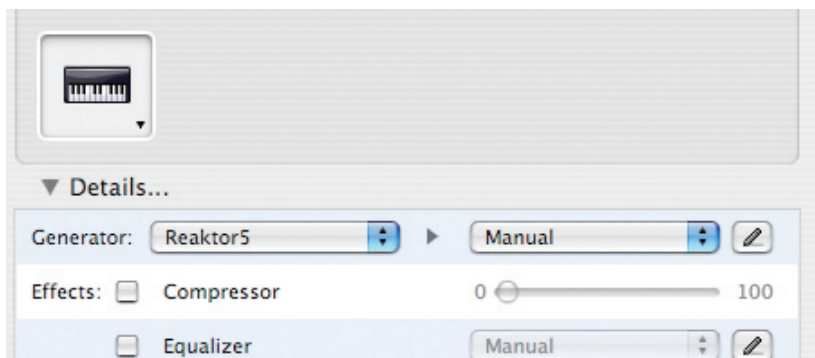
ライブラリーからインストールメントを読み込むと、キーボード・コントローラー側から MIDI 信号を送れば演奏できる状態になります。

REAKTOR 5 が生成する音声信号は、Digital Performer のミキサーを経由して直接サウンド・カードに送られます。MIDI 信号を受け取らない、あるいは音声信号を生成しない場合は、次の点を確認してみてください。

- ・ Digital Performer の **Studio** メニューで、**Midi Patch Through** がオンになっていること。
- ・ MIDI トラックの MIDI チャンネルが、インストゥルメントの受信チャンネルと一致していること。
- ・ インストゥルメント・トラック出力が正しく設定されていること。
- ・ サウンド・カードを Digital Performer 用に正しく設定していること。
(詳しくは Digital Performer のマニュアルも参照してください)

7.4.3. Garage Band でのプラグ・インの使い方

- ・ Garage Band を起動します。
- ・ 「+」 ボタンを押して、新たに「ソフトウェア・インストゥルメント」トラックを作成してください。表示に使うアイコンもここで選択できます。
- ・ インストゥルメント・トラックのアイコンをダブル・クリックするか、「I」アイコンを押して、トラック情報を表示します。
- ・ 情報ウィンドウで、「インストゥルメント」アイコンのすぐ下にある ▼ **Details...** を押すと、トラック設定欄が現れます。
- ・ **Generator** ドロップ・ダウン・メニューに並んでいる Audio Unit プラグ・インの中から、**REAKTOR 5** を選択してください。
- ・ **Manual** ドロップ・ダウン・メニューの隣にある「鉛筆」アイコンを押すと、REAKTOR 5 の設定編集画面が現れます。
- ・ これで、外づけ MIDI キーボードで REAKTOR 5 を演奏できるようになりました。



7.5. DXi 2 プラグ・イン

DXi は Microsoft DirectX 技術にもとづくプラグ・イン形式です。

7.5.1. Sonar 4 でのプラグ・インの使い方

- ・ Sonar を起動してください。
- ・ Synth Rack 画面で **REAKTOR 5 DXi 2** を選択してください。



Synth Rack 画面で DXi 2 を選択している様子

- ・ MIDIトラックから DXi 2 プラグ・インに信号が流れるようにします。
Out ドロップ・ダウン・リストで **REAKTOR 5** を選択してください。



MIDIトラックを DXi プラグ・インに割り当てている様子

ライブラリーからインストールを読み込むと、キーボード・コントローラー側から MIDI 信号を送れば演奏できる状態になります。REAKTOR 5 が生成する音声信号は、Sonar ミキサーを経由して直接サ

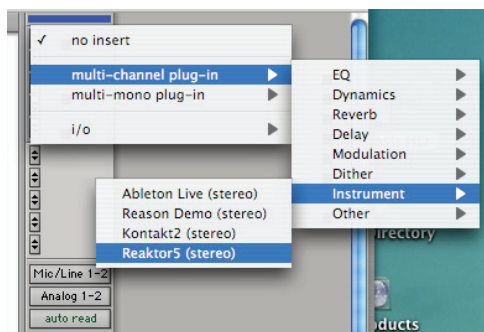
ウンド・カードに送られます。MIDI 信号を受け取らない、あるいは音声信号を生成しない場合は、次の点を確認してみてください。

- ・ Sonar 側の「Studio」メニューで「MIDI Patch Through」が有効になっていること。
- ・ MIDIトラックの MIDI チャンネルが、インストゥルメントの受信チャンネルと一致していること。
- ・ インストゥルメント・トラック出力が正しく設定されていること。
- ・ サウンド・カードを Sonar 用に正しく設定していること。
(詳しくは Sonar のマニュアルも参照してください)

7.6. Pro Tools 6.x での REAKTOR RTAS の使い方

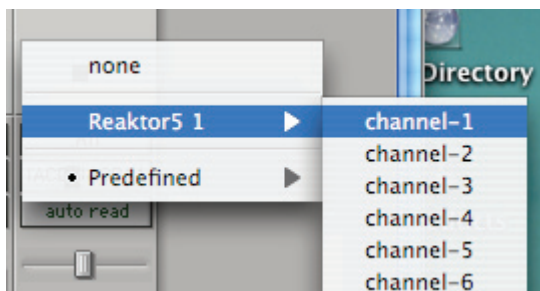
これは TDM ハードウェアの有無と関係なく ProTools 用のプラグ・インを使えるようにするためのプロトコルです。Mac OS 用、Windows 用の両方があり、豊富な機能が備わっています。必要な計算処理はすべてホスト・アプリケーション側で実行するようになっています。

- ・ Pro Tools を起動してください。
- ・ Auxトラック・ファイルを新たに作成してください。
- ・ 同様に、MIDIトラックを作成しておきます。
- ・ チャンネル・ミキサー画面を開き、ミキシング設定を表示してください。
- ・ Aux チャンネルの最上段にある濃灰色の部分で RTAS の制御区画です。先頭の空きスロットをクリックすると、使用可能な RTAS プラグ・インが列挙されたドロップ・ダウン・メニューが開きます。
- ・ メニューから **REAKTOR 5** を選択してください。



- ・ さらにこのサブ・メニューから、先に作成した MIDI チャンネル番号を選択してください。

- ・ 出力スロットで、REAKTOR の適当なチャンネルを選択します。



MIDIトラックを録音可能な状態にすると、MIDI キーボードで REAKTOR を演奏できるようになります。

重要：REAKTOR で作ったアンサンブルに複数の出力がある場合に、Pro Tools 6.7 以降でこれを使うためには、Pro Tools セッションにステレオ / モノラルの補助トラックを追加する必要があります。この補助トラックの入力源としてプラグ・インを割り当て、ドロップ・ダウン・メニューから該当する出力を選択してください。

(詳しくは Pro Tools のマニュアルも参照してください)

8. OSC(Open Sound Control)

OSC は、コンピューターやシンセサイザー、マルチメディア機器間で、情報をやり取りするためのプロトコルです。仕様が公開されており、特定のネットワークの機能に依存していない点が特徴です。MIDI と比べて信頼性が高く、使い勝手も向上しています。同一コンピューター上、あるいはネットワークでつながったコンピューター間で、さまざまな音楽アプリケーションを連携させたい場合に有用です。MIDI は規格に定義されたデータ（ノート・オン / オフ、ピッチベンドなど）しか扱えませんが、OSC を使えば、アプリケーションごとに独自のデータを階層化し、動的に管理することができます。

OSC はどんな通信基盤上でも動作し、もちろん TCP/IP ベースの LAN やインターネットでも使えます。ネットワークで多少の遅延が生じたり、その遅延時間が変動したりしても、時刻タグをつけてメッセージを束ねるしくみがあるので、曲の再生その他に影響はありません。また、さまざまなパラメーター型に対応しているので、今後 REAKTOR にも機能を組み込んでいく予定です。

8.1. 適用分野

REAKTOR の OSC 対応機能を活用すると、次のような使い方ができます。

- ・ インターネットを活用した国際分業体制による音楽作り。
- ・ 数十台規模のコンピューターを接続して音声データを共有。
- ・ 複数台のコンピューターによる分散処理。
- ・ 1 台のコンピューター上で複数の音楽ソフトウェアを動かして連携処理。

現行版では、コンピューター間でイベント・データをやり取りすることはできますが、オーディオ・データには対応していません。さらに、Ethernet 用の NIC(ネットワーク・インターフェイス・カード) を装着し、TCP/UDP/IP のプロトコル・スタックを組み込んでおく必要があります。

8.2. OSC システムの設定

OSC Setup

OSC
☒ Activate

LOCAL IP ADDRESS: 127.0.0.1
LOCAL PORT: 10000
LOCAL IDENTIFIER: Reaktor5-1
Apply

CLOCK SYNC
☐ Master ☒ Off
NETWORK DLY (MS): 0.000

TIME SYNC
☐ Master ☒ ok
TIME OFFSET (MS): 0

Identifier	IP Address	Port
------------	------------	------

Scan
Edit
Delete

IDENTIFIER: REMOTE IP ADDR.: REMOTE PORT: 0
Apply

OSC MESSAGE

MONITOR OPTIONS: select

OSC MONITOR

OK Cancel

OSC Setup ウィンドウ

OSC(Open Sound Control) に関する設定は、**System > OSC Settings** コマンドで、**OSC Setup** ウィンドウを開いて行います。TCP/IP を始めとする各種のネットワーク通信プロトコルを使い、外付け機器や他のアプリケーションと通信できます。

OSC の有効 / 無効の切り替え

OSC による通信は、**OSC Setup** ウィンドウの左上にある **Activate** ボタンで、有効 / 無効を切り替えることができます。OSC 通信は、REAKTOR のオーディオ処理が有効になっている場合に限り使えます。したがって、オーディオ・カードを装着するか、コンピューターに組み込みのオーディオ処理機能を有効にする必要があります。**Activate** ボタンの設定は大域的で、REAKTOR を複数起動している場合でも共通に適用されます。

OSC の識別名

OSC Setup ウィンドウの上部にはほかにも、**LOCAL IP ADDRESS**、**LOCAL IDENTIFIER**、**LOCAL PORT** を表示 / 設定する欄があります。この設定も大域的で、REAKTOR の各セッションに、共通に適用されます。

- ・ **LOCAL IP ADDRESS**: コンピューターに割り当てられた IP アドレスを表示します。自動認識され、編集はできません。
- ・ **LOCAL IDENTIFIER**: 他の OSC クライアントと区別するための識別名です。任意の文字列を指定できます。
- ・ **LOCAL PORT**: サブネットワークの識別子です。ネットワーク上にある他の OSC クライアントは、この識別子によって互いを認識します (後述の **Scan** ボタンも参照)。10000 ~ 10015 番の範囲で選択してください。
- ・ **Apply**: 以上の設定を修正した場合は、**Apply** ボタンを押して変更を有効にしてください。

OSC の同期処理

OSC クライアント間の同期に関する設定です。マスター側からクロック信号を供給する「クロック同期」方式と、スレイブ側から定期的にタイム・スタンプを問い合わせて時刻を調整する「時刻同期」方式があります。クロック同期方式のしくみは、MIDI クロックによる同期と同じです。

- ・ **CLOCK SYNC-Master**: オンにすると、「クロック同期」方式のマスターとして、メンバー・リスト (後述) に載っている他の OSC クライアントに OSC クロック信号を送るようになります。

- ・ **TIME SYNC-Master**: オンにすると、「時刻同期」方式のマスターとして、時刻の基準となります。
- ・ **SELECT MASTER**: **Master** がオフ、すなわちスレイブ側の場合に、マスターを選択します。クロック同期の場合は **Clock Sync**、時刻同期の場合は同期元 (ポーリング先) クライアントの識別名を指定してください。
- ・ (同期を表す LED): **CLOCK SYNC**、**TIME SYNC** の各チェックボックスの右側にある LED 表示器は、同期信号を送受信する時に点灯します。
- ・ **SYNC MESSAGES**: 同期を取る際にエラーが発生すればこの欄に表示されます。
- ・ **TIME OFFSET (MS)**: 時刻同期方式の場合に、OSC メッセージを送信してから、相手方が受信するまでの時間を表します。例えば「1000」と設定すれば、実際に相手クライアントが受信するのは 1000 ミリ秒後ということになります。

OSC メンバー・リスト

接続を確立するべき、相手先 OSC クライアントをすべて列挙したものです。

リストの各項目は必要に応じて編集 / 削除できます。編集したい項目を選んで **Edit** ボタンを押してください。なお、修正した場合は **Apply** ボタンを押して、変更を有効にする必要があります。

OSC クライアントを削除したい場合は、選択して **Delete** ボタンを押してください。

Scan ボタンを押すと、サブ・ネットワーク内にある OSC クライアントを自動的に検出し、リストに表示します。但し、次の条件を満たすクライアントが対象です。

- ・ 同一サブネット内にあること。
- ・ REAKTOR が稼動している (オーディオ・エンジンが有効になっている) こと。
- ・ **OSC Setup** ウィンドウでの **Activate** がオンになっていること。
- ・ **LOCAL PORT** として 10000 ~ 10015 の範囲の番号を指定してあること。

同じサブネットにはない 2 台のコンピューターを (例えばインターネット経由で) 接続したい場合は、メンバー・リストの下にある

IDENTIFIER、**REMOTE IP ADDR.**、**REMOTE PORT** 欄に相手方コンピュータの識別名、IP アドレス、ポート番号を入力し、**Apply** ボタンを押してください。

OSC の監視

OSC Setup ウィンドウの一番下は、OSC の動作を監視するための操作領域になっています。

- ・ **OSC MESSAGE:** 他の OSC クライアントにメッセージを送るために使います。OSC 接続のテストや、チャット用に利用できます。メンバー・リストから送信相手を選択し、メッセージを入力したあと **return** キーを押してください。
- ・ **OSC MONITOR:** 受け取った OSC メッセージを表示します。
- ・ **MONITOR OPTIONS:** 監視用ウィンドウに関する機能を設定します。

9. REAKTOR 最初の一步

この章では、REAKTOR の基本操作と機能、アンサンブル / インストゥルメントの作成方法について、実例を紹介しながら解説します。

REAKTOR は誰でも簡単に使えるとか、ほんの数分でハードウェア・シンセサイザーと同等の機能を構築できるとかいった、かりそめの希望を抱かせるようなことは言いません。実際には、REAKTOR は非常に複雑なプログラムであり、高度な機能を実現するためのさまざまなしくみが組み込まれています。始めの一步として、この章を読みながら基本的な使い方を学び取ってください。これを怠ると後で苦勞することになるでしょう。

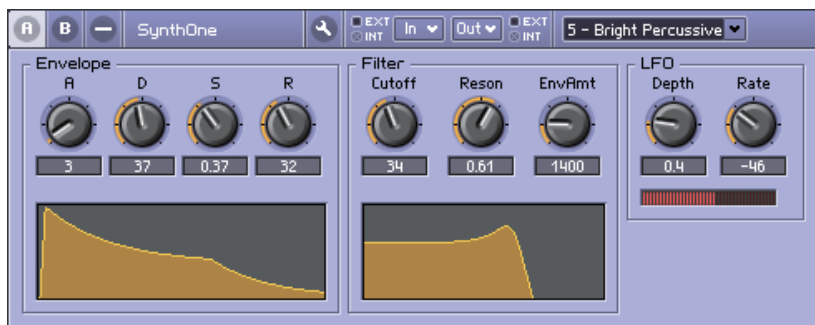
しかし心配はいりません。確かに REAKTOR には高度な機能が満載ですが、最初からすべてを理解している必要はないのです。この章では、ソフトウェアで実現されたいくつかのインストゥルメントを使って曲作りを試みますが、音声合成の理論や処理方式などについては分からなくても構いません。あらかじめ用意されたライブラリーの使い方だけ理解してください。

9.1. 作成済みインストゥルメントの演奏

最初に、MIDI コントローラー (マスター・キーボードや MIDI ワークステーション) が、コンピューターの MIDI 入力に正しく接続されているかどうか確認してください。 **Audio Setup** 画面の **MIDI** タブで、 **Input Interface** リスト上の該当する入力ポートを **on** にしておく必要があります (第 6 章「スタンドアローン版の REAKTOR 5」も参照)。MIDI 送信チャンネルは 1 にします。

コンピューター側キーボードの QWERTY キーで代用することも可能です。付録「音符に対応するコンピューター・キーボード上のキー」も参照してください。

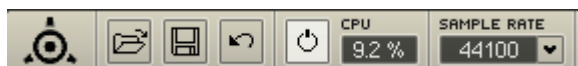
SynthOne



SynthOne の制御パネル

REAKTOR のインストール先、ライブラリー・フォルダーの「Tutorial Ensembles」以下に、「SynthOne.ens」というファイルがあります。**File > Open...** コマンドを実行してこのファイルを選択するか、REAKTOR のブラウザー上でファイルをつかみ、主ウィンドウまでドラッグしてください。

まず、主ツールバーの機能を見てみましょう。



Run/Stop Audio ボタンがオンで、その隣に CPU に対する負荷が表示されているでしょうか。この状態ならば SynthOne が動作していますが、そうでなければ **Run/Stop Audio** ボタンを押して、シンセサイザーを起動してください。

ここで、**CPU** 欄の表示が「Over」となっている、あるいは「Processor Overload!」という警告が現れている場合は、いったんオーディオ処理を停止し、設定を変えて負荷を軽減しなければなりません。インストールメント・ヘッダーの **VOICES** 設定（声部数）が 6 になっているのを減らすか、ツールバーの **SAMPLE RATE** 設定を 44100Hz より小さい値にしてください。また、**Out** レベル・メーターが赤く点灯している場合は、サウンド・カードが過負荷になっているので、**Main** フェーダーでアンサンブル全体の音量を絞ってください。

MIDI キーボードのキーを叩くのに応じて、SynthOne インストゥルメント・ヘッダーにある、**In** 側のランプが赤く点灯するはずです。



これは SynthOne が MIDI データを受け取っていることを表します。**INT** は内蔵 MIDI 機器、**EXT** は外付け MIDI 機器からの信号があると点灯します。

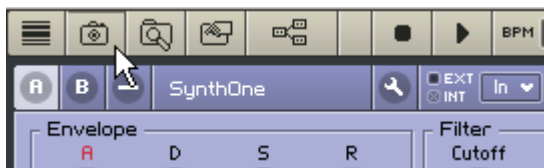
SynthOne は単純な 6 声部アナログ・シンセサイザーを真似たものです。鋸波発振器、24dB/oct. のロー・パス・フィルター、ピッチを変調するための LFO、ADSR エンベロープ生成器から成ります。エンベロープは、鋸波の振幅やフィルターのカットオフ周波数を変調するために使います。

SynthOne の制御パネルには、音の響きを調整するためのつまみ類がいくつか並んでいます。左から順に、エンベロープに影響を与える **A**(アタック)、**D**(デケイ)、**S**(サスティン)、**R**(リリース) の各つまみ、フィルター特性を調節する **Cutoff**(カットオフ周波数)、**Reson**(レゾナンス; カットオフ周波数付近の帯域を強調する度合い)、**EnvAmt**(フィルター特性をエンベロープ曲線によって変調する度合い)、さらに LFO 関係のつまみとして、**Depth**(発振周波数を変調する度合い)、**Rate**(発振周波数) があります。

少しでもシンセサイザーを使ったことがあれば、SynthOne もそれ程抵抗なく使ってみることができるでしょう。一方、これが初めて使うシンセサイザーである場合は、この機会にいろいろ調節して、響きが変わるか実験してみてください。ここに並んでいるつまみ類は、いずれも音声合成の基本的なパラメーターを調節するものです。いくらいっても、何かが壊れてしまうような心配はありません。

気に入った響きが見つかったら、そのパラメーター設定を保存しておくといでしょう。次のように操作してください。

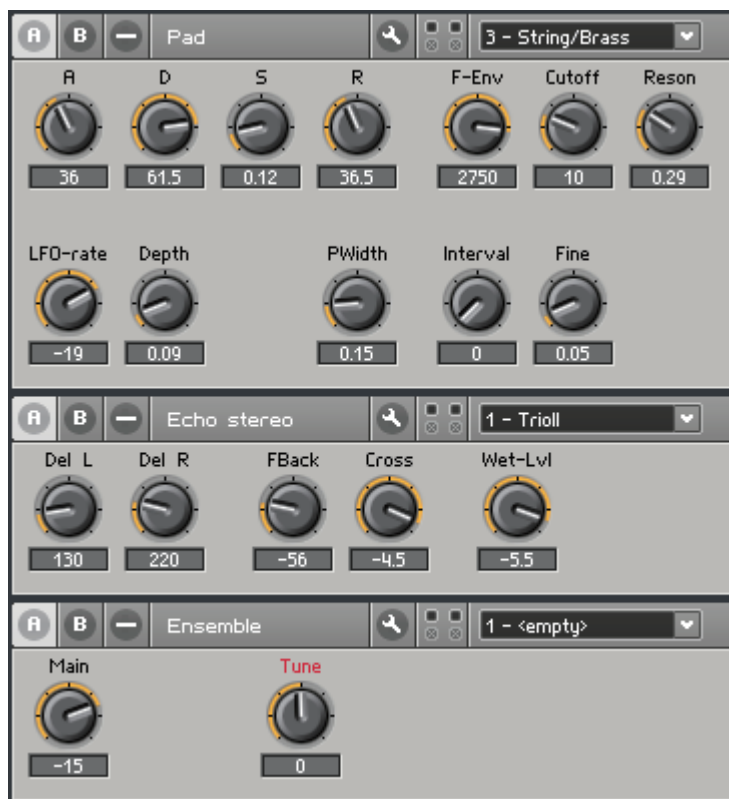
- ・スナップショット・ウィンドウを開きます。ツールバーにある、カメラをかたどったボタンを押してください。



これはスナップショットの保存、改名、削除などに使う画面です。他のシンセサイザーでは、「パッチ」、「プリセット」、「プログラム」などと呼ばれているものに相当します。

- ・ **APPEND** ボタンを 2 回押すと、現時点のつまみ類の設定が保存されます (1 回押すとボタンが点灯し、もう 1 度押すと元に戻ります)。スナップショットの保存枠 (スロット) が並んでいますが、最初に見つかる空きスロットが保存先になります。
- ・ このスナップショットには仮の名前がついています。名前の箇所をダブル・クリックし、新しい名前を入力して **return** キーを押してください。

Pad Echo



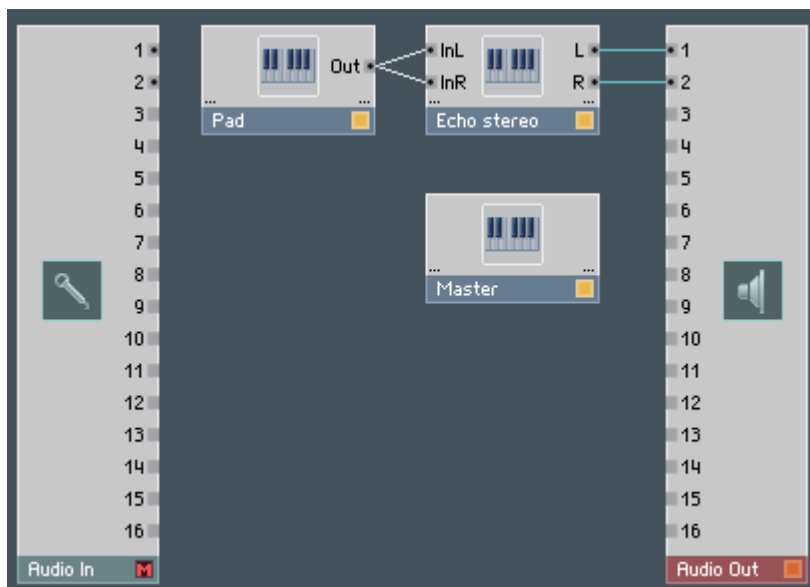
次に「Padecho.ens」というアンサンブルを見てみましょう。これも SynthOne と同じフォルダーにあります。

REAKTOR でこのアンサンブルを開こうとすると、前節で SynthOne にいろいろ変更を加えたので、保存するかどうか確認する画面が現れます。特に必要なければ **No** を押してください。

「Pad」、「Echo stereo」、「Ensemble」という標題のついた、3つの制御パネルが現れます。アンサンブル制御パネルのツールバーにある、**Structure** ボタンを押してください。(モジュールを表す)3つの小さな箱を線で結んだ形のアイコンがついたボタンです。するとアンサンブルのストラクチャーを表示するウィンドウが開きます。



「アンサンブル」(Ensemble) とは、REAKTOR のインストゥルメントの中でも最上位階層に位置するものを言います。このストラクチャー・ウィンドウでは、全体構成を概観できます。「Padecho」の場合、シンセサイザーの制御部に当たる「Pad」、ステレオ・ディレイ・エフェクトである「Echo Stereo」、全体の音量調整 / 調律を行う「Master」の、3つの要素から構成されています。



「Pad」の出力は、「Echo Stereo」の左右の入力につながっています。さらに「Echo Stereo」の出力は、**Audio Out** モジュールの入力 (上から2つ) に接続されています。

Audio Out はどのアンサンブルにも必要なモジュールで、生成した信号を REAKTOR の外部に送り出す働きがあります。一般にはサウンド・カードのオーディオ出力に当たりますが、プラグ・インの形で他のソフトウェアに信号を供給することも可能です。一方、これと反対の働きがある、**Audio In** というモジュールもあります。サウンド・カードのオーディオ入力に当たりますが、同様にプラグ・インの形で他のソフトウェアから信号を受け取ることもできます。これもすべてのアンサンブルに必要ですが、Padecho ではオーディオ入力を使わないので、ミュート (無音) 状態になっています。

以上からも分かるように、アンサンブルは複数のインストゥルメントから構成します。したがって、単なる音声合成以外の機能も組み込むことが可能で、例えば「Echo Stereo」には、音声信号を加工する (エフェクト) 働きがあります。

ストラクチャー・ウィンドウは、背景の濃灰色部分をダブル・クリックすると制御パネルに切り替わります。逆に制御パネルは、背景の黒い部分をダブル・クリックするとストラクチャー・ウィンドウになります。

アンサンブルを使って演奏する際、ストラクチャー・ウィンドウを開いたままにしておいても構いませんが、自由に調節できるつまみ類があるわけではないのであまり役には立たないでしょう。

「Master」にはノブが2つついています。すなわち、全体の音量調整に使う **Main** ノブと、調律用の **Tune** ノブです。

同様に、「Pad」や「Stereo Echo」にも調整つまみ類がいくつかついています。

「Padecho」アンサンブルで実際に曲作りを試みるのに先立ち、そのストラクチャー（構成）について若干説明しておきましょう。シンセサイザー「Pad」には、パルス波を生成する発振器が2つついています。第2発振器の周波数は、第1発振器を基準とした相対値で調整します。**Interval** ノブで大雑把に合わせ、**Fine** ノブで微調整してください。どちらの発振器もパルス幅は **PWidth** で調整しますが、LFO（低周波発振器）で変調をかけ、パルス幅が揺れるようにすることも可能です。変調の速度と深さは **LFO-rate** および **Depth** で設定します。フィルターや振幅に影響するADSRエンベロープ、画面右側のノブで調整するフィルターについては、SynthOneと同様です。

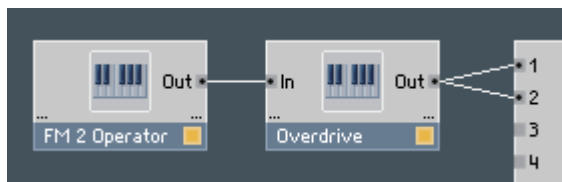
エフェクト「Echo Stereo」には、左右のチャンネルに対応して2本のディレイ・ライン（遅延線）があります。遅延時間は **Del L** および **Del R** で独立に調整できます。これを0とすれば遅延が生じません。エコーを繰り返す回数は **FBack** ノブ、**Cross** ノブで設定します。左右のチャンネルの信号がある比率で減衰して戻ってくることによりエコー効果が生まれるわけですが、同じチャンネルに戻ってくる信号の減衰率が **FBack**、反対側のチャンネルに戻ってくる信号の減衰率が **Cross** です。最後に **Wet-Lvl** は、元信号をディレイ・ラインに加える割合を表し、これがエフェクトの強さに反映されます。

FM Overdrive



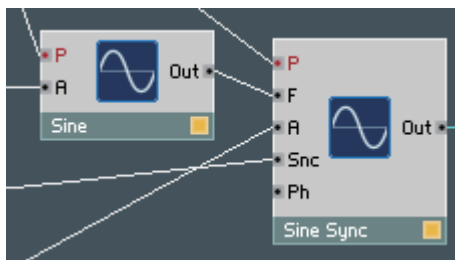
次は FM 方式の音声合成の例で、やはり同じ「Tutorial Ensembles」以下にある、「FM-Overdrive.ens」というファイルです。

ストラクチャーを見ると分かるように、シンセサイザー本体である「FM 2 Operator」と、歪み系のエフェクトである「Overdrive」を組み合わせた恰好になっています。



REAKTOR には減算方式以外にも多様な音声合成方式が組み込まれていますが、「FM 2 Operator」はその特徴をうまく活かしています。ここでは、Yamaha DX シリーズに採用されたことで広く普及した、FM(周波数変調)方式を使っています。

DX7 のような 6 オペレーター方式や、その普及版である DX シリーズに使われている 4 オペレーター方式ではなく、より簡潔な 2 オペレーター方式で構成しました。



どちらのオペレーターも、その実体は正弦波発振器です。一方は被変調波で、基底となるピッチがこれで決まります。もう一方は変調波で、被変調波の周波数を変動させることにより音色を調整します。

FM ノブを調整しながら MIDI インストゥルメントを演奏し、響きがどのように変わるか聴いてみてください。



FM ノブを右に回すにつれて、金管系の輝かしい響きが加わっていくのが分かります。このノブは、基底となる信号 (被変調波) を変調する度合い (変調強度) を調整しているのです。

次に **Interval** ノブを調節してみましょう。少し動かすだけで、音が大きく変わるはずです。隣の **Detune** ノブは微調整に使います。

時間の経過とともに音量や音色を変えていくために、ごく単純なエンベロープ生成器を使っています。被変調波の音量を調整するエンベロープは、デケイとリリースの、2つのパラメーターしかありません。変調波のエンベロープはもっと単純で、**Mod-D** ノブでデケイ時間を調整できるだけです。

以上のような原理を頭に入れておけば、比較的簡単に欲しい音を作り出せるはずです。

次に「Overdrive」について簡単に見てみましょう。これは、実際の楽器を演奏する際に現れる、ある種のきしみ音を出そうとするものです。

さまざまな設定を、スナップショットを取って切り替えながら試してみたと、以下の説明を読み返せばよく理解できるでしょう。

Drive は歪みを加える回路に送る信号量を表します。最終的に加わる歪みの大きさが、この信号量によって決まります。**Asym** ノブは倍音構成の制御用で、真空管アンプのような暖かみのある音色を作り出すことができます。これに続くフィルター部では、**Freq** ノブでカットオフ周波数、**Emph** ノブでカットオフ周波数付近の帯域を強調する度合いを調整できます。最後に **Volume** ノブで音量を調節します。

16 ステップ・シーケンサー + ベースライン



次に「Squnc16*.ens」アンサンブルを見てみましょう。これも同じ「Tutorial Ensembles」フォルダーにあります。



ストラクチャーを見ると分かるように、16 ステップ・シーケンサー「Sequencer16」を使って「Bassline」を制御する恰好になっています。「Bassline」は TB-303 という仮想アナログ・シンセサイザーをディジ

タル的に再現したインストゥルメントです。この出力を「Auto-Panner」でステレオ信号にします。

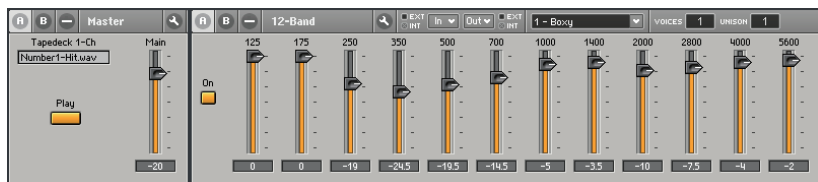
「Sequencer16」の制御パネルを開いて **Run** ボタンを押すと、あるパターンの音が繰り返し聴こえてきます (**Run** ボタンで再生 / 停止を切り替え)。16のステップごとに、上段のフェーダーでピッチ、下段のフェーダーで音量レベルを設定できます。テンポは **BPM** ノブ (**Run** ボタンの隣)、各ステップの長さは **Length** ノブで調節します。

Reset ボタン (**Run** のすぐ下) を押すと先頭のステップに戻ります。シーケンサーの動作中にこのボタンを押すと、一定幅ずれたパターンになります。一方、停止中に押した場合は、(前回停止した時の位置にかかわらず、) 次に **Run** を押したとき、先頭から再生が始まるようになります。

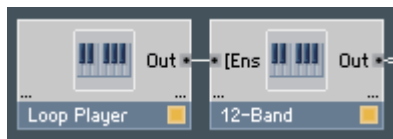
「Bassline」インストゥルメントは先にも述べたように TB-303 を再現したのですが、これを使ったことがなくても、つまみ類がほんの少ししかないので迷うことはないでしょう。自由に調節し、音を聴き比べてみてください。

このアンサンプルの出力は、ステレオ場上进行ったり来たりしています。これを実現するのが「Auto Panner」の役割です。左右のチャンネルを行き来する幅は **Amount** ノブ、速度は **Rate** ノブで調節します。

サンプル・ループ・プレーヤー



最後の例として「Wav-play*.ens」を取り上げましょう。これも「Tutorial Ensembles」フォルダーにあります。



これは「Loop Player」と、12 帯域イコライザー「12-Band」の、2つのインストゥルメントを組み合わせたものです。まず、既存のサン

プルを読み込み、再生してみてください。制御パネル上の「untitled*.wav」と表示されたスロット上でコンテキスト・メニューを開き、**Load Audio in Tapedeck...**を実行します。すると**Open Audio File**ダイアログ・ボックスが開くので、ハード・ディスク中の適当な WAV/AIF ファイルを選択し、**Open**を押してください。**Play** ボタンで再生が始まります。音が聴こえない、あるいは正しくループしない場合は、**Play** ボタンを 2 回押して、初期化し直すと正常になるはずです。

しかしこのアンサンプルの目玉となる機能は「12-Band」エフェクトにあります。制御パネルは一般的なグラフィック・イコライザーと同様、周波数帯域ごとにフェーダーで調整するようになっています。しかしここで使われているのはフィルター・バンクなので、イコライザーよりも思い切った調整が可能です。各フェーダーには、帯域 (Hz 単位) を表す番号がついています。サンプルを繰り返し再生しながら調整、比較してみるとよいでしょう。響きを変えるだけにとどまらず、特定の音域をほぼ完全に消してしまうなど、音楽的な特性そのものすら変えてしまうことができるのです。

9.2. 基本的なシンセサイザーの構築

前節で紹介した例を始めとして、作成済みのインストゥルメントやエフェクトなどが、ライブラリーとして多数揃っています。しかし REAKTOR を使う醍醐味は、何といっても独自のインストゥルメントを設計、構築することでしょう。基本から順に進めていけば、決して難しいことはありません。

ここでは伝統的なアナログ・シンセサイザーを取り上げます。減算方式、すなわち、倍音成分を豊富に含む信号をまず生成しておき、フィルターで不要な成分を「減算」していくことにより、望みの音を作り出す方法です。

準備

シンセサイザーの構築には「マクロ」を積極的に使っていくことにします。

注意：REAKTOR には「基本マクロ」と「コア・マクロ」の2種類があり、基本レベルとコア・レベルにそれぞれ対応しています。ここでは、単にマクロと言えば「基本マクロ」を指します。コア・マクロについては「REAKTOR 5 CORE チュートリアル / オペレーション・マニュアル」を参照してください。

REAKTOR で言う「マクロ」とは、あるストラクチャーを機能部品としてまとめたもののことです。このしくみにより、複雑なストラクチャーが構築しやすくなるのはもちろんですが、構成要素の配置が分かりやすくなる、という効能もあります。最初から多様なマクロがライブラリーとして付属しているので、インストゥルメントの構築におおいに役立ててください。

まず、主ツールバーの **Run/Stop Audio** ボタンをオフにして、作成途中に突然音が鳴り出すことがないようにします。



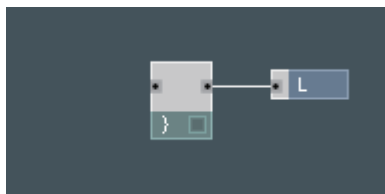
シンセサイザーを構築していく「作業台」を用意しましょう。**File > New Ensemble** コマンドを実行すると、次のような空のストラクチャーができます。これまでも出てきた **Audio In**、**Audio Out** モジュールのほか、**Instrument** および **Master** というインストゥルメントがあります。



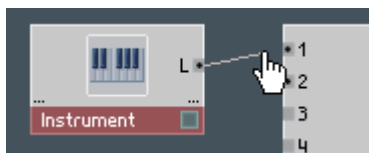
しかし今回は何もないところからアンサンブルを構築するので、**Instrument** は削除しておきます。**Master** の方はいずれにしても必要になるので、そのまま残します。**Master** の制御パネルには、全体的な音量調整 / 調律のための **Level**、**Tune** ノブがついているからです。

まず、シンセサイザーを組み込んでいく「外枠」が必要です。そこで、中身が空のインストゥルメントを、ライブラリーから追加することにしします。アンサンブル・ストラクチャー・ウィンドウの、何もない背景部分でコンテキスト・メニューを開き、**Insert Instrument > New-2In2Out** コマンドを実行してください。**Instrument** という名前の、空のインストゥルメントが現れます。

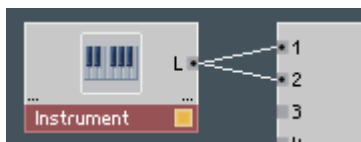
これをダブル・クリックして **Instrument** のストラクチャー編集画面に切り替え、**L** という出力端子と、その前段にある **Audio Voice Combiner** モジュール(名前として「J」と表示されているもの)を除いて、すべて削除してください。



ストラクチャー編集画面の背景部分をダブル・クリックすると、アンサンプルのストラクチャー表示に戻ります。このように、ストラクチャーの上位階層に戻るには、背景部分をダブル・クリックするのが最も手軽な方法です。**Instrument** の **L** 出力ポート上をクリックし、そのままドラッグして、**Audio Out** モジュールの 1 番入力ポート上で放してください。



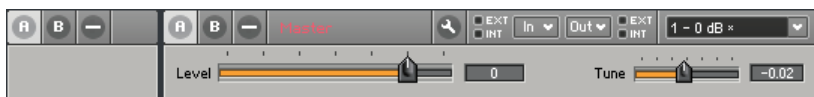
Instrument と **Audio Out** の間が線が結ばれたはずですが。これが仮想的な結線を作る手順です。同じ手順で、**L** 出力ポートから **Audio Out** の 2 番入力ポートにも結線を作ってください。こうしておけば、後で左右の両方のチャンネルから音が聴こえるようになります。



コンポーネントの選択

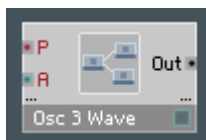
シンセサイザーを作るのですから、発振器をいくつか用意する必要があります。その出力信号をフィルターで加工した後、エンベロープに沿って音量を制御します。したがって、信号経路に沿って、上記の手順で各コンポーネントを結ぶことになります。

アンサンプル制御パネル・ウィンドウを見てみると、その中には **Master** および **Instrument** の制御パネルがあります (ストラクチャーのことではないので注意)。



とは言っても **Instrument** の方は、まだストラクチャーを構築していないので、制御つまみ類は何もありません。一方 **Master** の制御パネルには、**Level**、**Tune** というフェーダーがあります。

次に、**Instrument** のストラクチャー編集画面を開きます。ここに発振器その他のモジュールを追加していきます。コンテキスト・メニューを開き、**Macro > Building Blocks > Oscillators > Osc (pls, saw, tri)** を実行してください。すると **Osc 3 Wave** というマクロが追加されます。

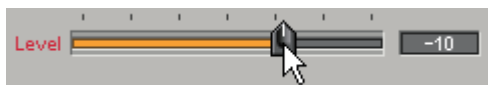


制御パネルにも **Osc 3 Wave** 用の制御つまみ類が追加されています。



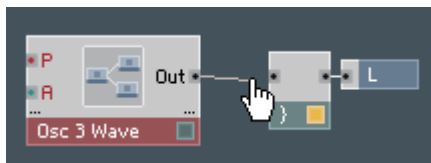
これで完成に一步近づきました。

しかし先に進む前に、発振器が正常に動作するかどうか確認しておきましょう。**Master** 制御パネルの **Level** フェーダーを -10 程度にして、急に大きな音が出てしまわないようにしてください。

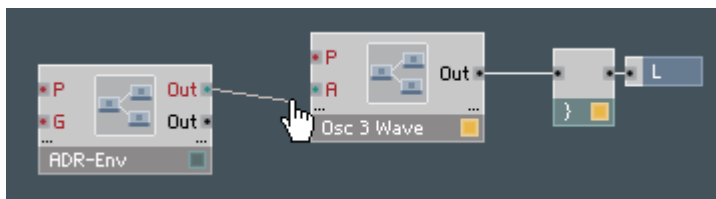


Instrument のストラクチャー編集画面で、**Osc 3 Wave** マクロの **Out** ポートと、**Audio Voice Combiner** モジュールの入力ポートとを、先ほどと同じ手順で接続します。**Audio Voice Combiner** には多

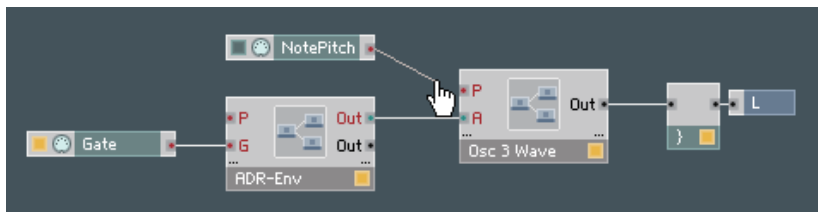
声信号を単声信号に変換する働きがあります。インストゥルメントの出力ポートは一般に単声なので、その前段には変換モジュールが欠かせません。



ここに、音量を変化させるための ADR エンベロープ生成器を追加しましょう。やはりコンテキスト・メニューから、**Macro > Building Blocks > Envelopes > ADR-Env** を実行します。そして、出力ポートのうち上側 (赤い方) と、**Osc 3 Wave** マクロの **A** 入力とを、結線で結びます。

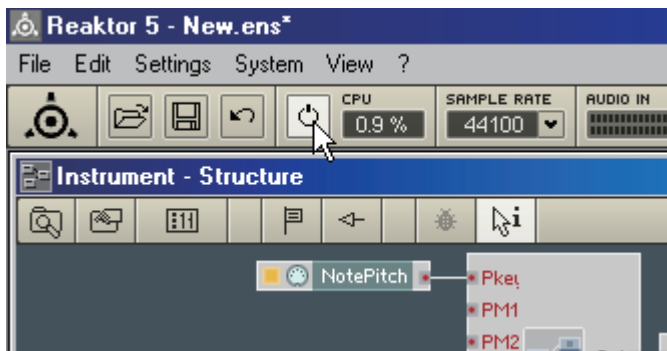


さらに、外部 MIDI 入力機器で操作できるよう、もう 2 つモジュールを追加します。1 つは **Note Pitch** モジュールで、**Built-In Module > MIDI In > Note Pitch** コマンドで追加し、その出力を **Osc 3 Wave** の **P** 入力につないでください。もう 1 つは **Gate** モジュール (**Built-In Module > MIDI In > Gate**) で、その出力を **ADR-Env** マクロの **G** 入力につなぎます。



主ツールバーの **Run/Stop Audio** ボタンをオンにし、MIDI キーボードで何か演奏してみてください。シンセサイザーの音が聴こえるはずです。聴こえない場合は結線を再確認し、アンサンプルをいったん保存し

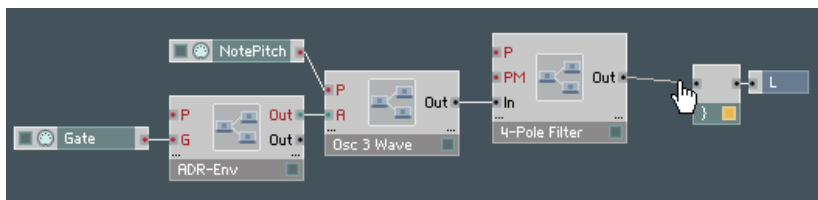
たあと読み込み直してみてください。これで一応音が出ることは確認できました。もっとも、発振器の音がじかに出ているだけなので、まともな音楽にはなりません。



次にフィルターを組み込みますが、その前に **Osc 3 Wave** と **Audio Voice Combiner** を結ぶ結線を切断しておきましょう。一方のポート上をクリックし、(結線がつかないときと同じように)ドラッグしてもう一方のポート上で放してください。別法として、結線上をクリックして選択し(線の色が変化)、**delete** キーを押しても構いません。

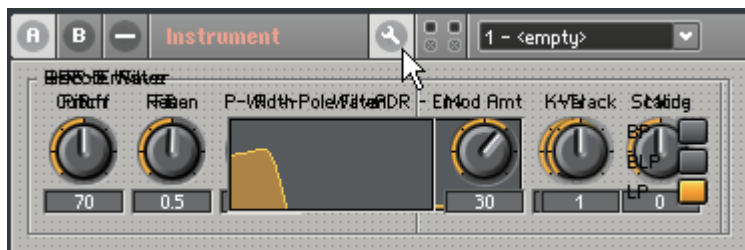


いよいよフィルターを追加します。これまでと同様に、コンテキスト・メニューから **Macro > Building Blocks > Filter > 4 Pole Filter (BP, BLP, LP)** を実行してください。Osc 3 Wave の Out ポートと 4 Pole Filter マクロの In ポートをつなぎ、さらに 4 Pole Filter の Out ポートと Audio Voice Combiner の入力ポートを結んで、発振信号が L ポートまで到達できるようにします。

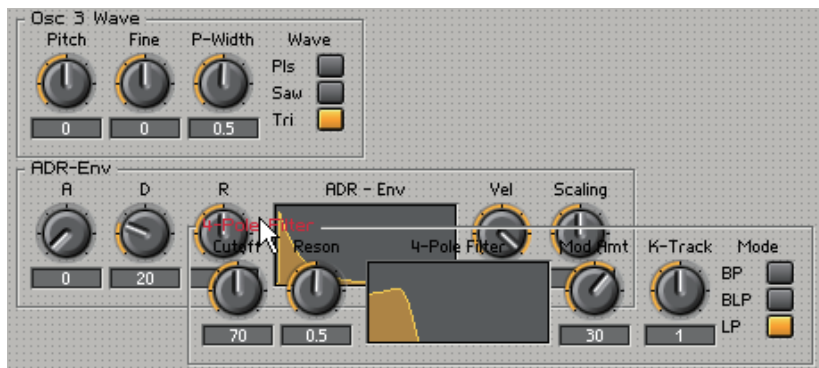


ここで制御パネルを見ると、フィルターの動作を制御するためのつまみ類が増えています。しかし発振器のつまみ類と重なり合って、まともに使えない状態になっているかも知れません。そこで次の手順で配置換えしてください。

1. インストゥルメント制御パネル・ヘッダーの **Lock/Unlock Panel** ボタン (スパーをかたどったアイコン) を押して、ロックを解除しておきます。



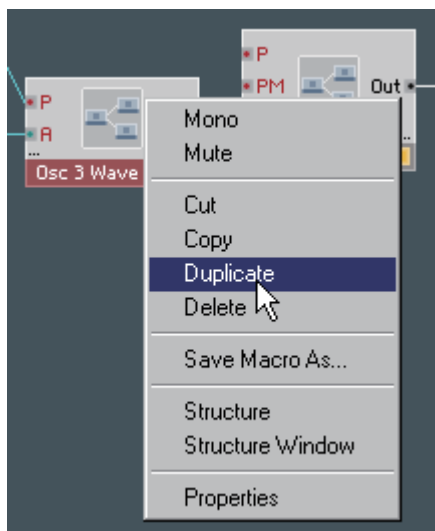
2. 各マクロに対応する枠の左上、タイトル部分をマウスでつかんでドラッグすると、自由に位置を調節できます。



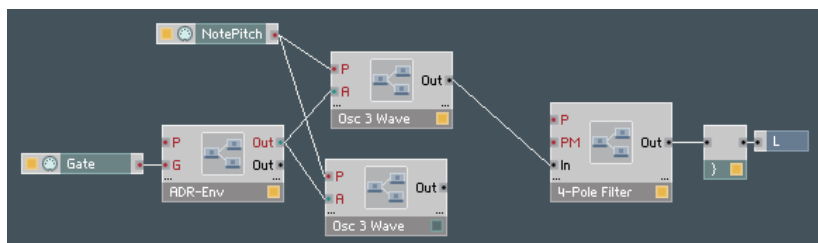
3. 配置を適切に調節できたら、もう一度 **Lock/Unlock Panel** ボタンを押してロックし直してください。次のように、**Osc 3 Wave**、**ADR-Env**、**4 Pole Filter** の各マクロに対応する区画を縦に並べるとよいでしょう。



今の段階では発振器が1つしかありませんが、単に2つ使うだけでも音に厚みを加えることができることはよく知られています。そこで、**Osc 3 Wave** 上でコンテキスト・メニューを開き、**Duplicate** コマンドを実行してください(この時ポート上をクリックしないよう注意)。

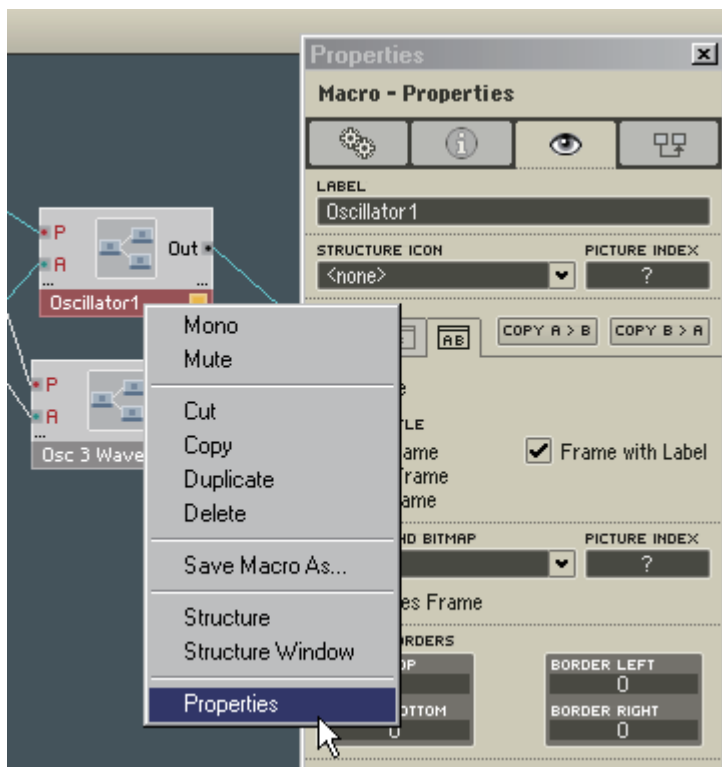


今回のように複雑なシンセサイザーを構築する場合、すっきりと分かりやすい設計をすることが特に大切です。ややもすると錯綜した配置になり、問題が起こったとき原因を見つけるのに苦労することになってしまいます。このあたりでストラクチャー・ウィンドウの配置を整理しておきましょう。**4 Pole Filter** マクロを、**Audio Voice Combiner** モジュールをはさんで **Out** ポートの左側に移動します。さらにその左側に、2つの **Osc 3 Wave** マクロを縦に並べて配置してください。



2つの発振器の名前が同じままだといささか混乱の元になるので、改名しておきましょう。上の方の **Osc 3 Wave** マクロ上でコンテキスト・メニューを開き、**Properties** コマンドを実行してください。するとプロパティ画面が開くので、その **LABEL** 欄に、例えば「Oscillator1」と入力します。もう一方の **Osc 3 Wave** マクロについても、同様の手

順で「Oscillator2」と改名してください。さらに、**4 Pole Filter** についても、単に「Filter」などと変更しても構いません。

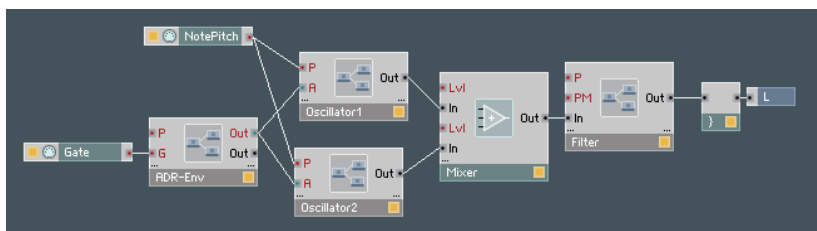


現状では、「Oscillator1」の発振信号しか、フィルターを介して音になって出力されていません。そこで「Oscillator2」の **Out** ポートとフィルターの **In** ポートを接続したいのですが、同じ入力ポートに2本以上の結線をつなぐことはできない、という問題があります。ハードウェア機器の入力端子と対比すれば、これは当然のことと言えるでしょう。そこで、両方の発振信号をひとつにまとめる（混ぜ合わせる）部品が必要です。この役割を担うのが **Mixer** モジュールです。

ストラクチャー・ウィンドウの背景部分でコンテキスト・メニューを開き、**Built-In Module > Signal Path > Amp/Mixer** を実行してください。



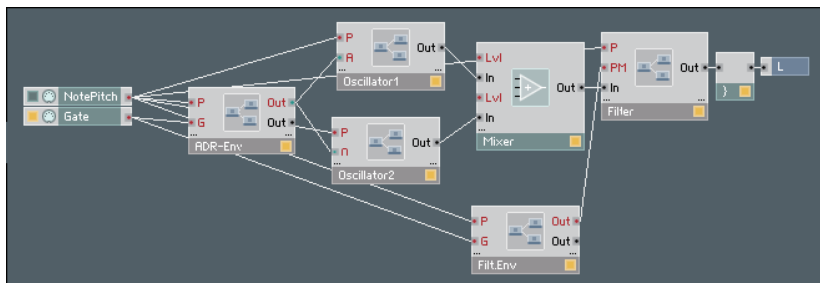
Mixer モジュールを発振器とフィルターの間に配置し、「Oscillator1」の出力を **Mixer** の **In** ポートに接続します。同様に「Oscillator2」の出力も **Mixer** につなぎたいのですが、**In** ポートはもう使ってしまったし、ここに2本以上の結線をつなぐことができないことは先ほどと同じです。しかしこのモジュールの場合、必要に応じて **In** ポートを増やせるので問題ありません。Windows ならば **Ctrl** キー、MacOS X ならば **command** キーを押しながら、「Oscillator2」の **Out** ポートをクリックし、そのままドラッグして **Mixer** の **In** ポート上で放してください。すると新たに **In** ポートができ、接続も終わっています。あとは今まで通り、**Mixer** の **Out** ポートとフィルターの **In** ポート、フィルターの **Out** ポートと **Audio Voice Combiner** の入力ポートを接続してください。



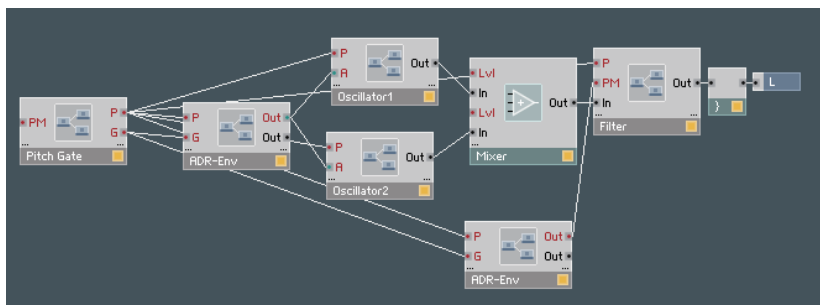
ここまで接続が終わると、各モジュールの状態を表す LED が点灯するので、ストラクチャーが動作可能であることが分かります。

なお、2つ目の発振器モジュール「Oscillator2」を作るには、**Duplicate** コマンドではなく、**Copy/Paste** 操作で複製するやり方も可能です。但しこの場合、**Note Pitch** および **ADR-Env** の出力を、「Oscillator1」と同様に、**P**、**A** 入力にそれぞれ接続する必要があります。

もう1つ ADR エンベロープ生成器を用意し、これでフィルターの動作を制御するようにすれば、音の響きがさらに豊かになります。**ADR-Env** マクロを **Duplicate** コマンドで複製し、その **Out** ポートのうち上の方(赤)を、フィルターの **PM** 入力に接続してください。新しいエンベロープ生成器の名前を「Filt.Env」と変更しておきましょう。



最後に、**Note Pitch**と**Gate**を、両方の機能を統合した**Pitch Gate**マクロに置き換えます。**Note Pitch**と**Gate**を削除し、代わりに**Pitch Gate**マクロ (Macro > Building Blocks > Pitch+Gate > Pitch + Gate) を追加します。その**P**出力を、2つの**ADR-Env**マクロ、**Oscillator 1**および**Oscillator 2**、**Filter**の**P**入力に接続してください。また、**G**出力は2つの**ADR-Env**マクロの**G**入力に接続します。



これできちんとした音の出るシンセサイザーが仕上がりました。MIDI信号を送ると、音高や音量を認識して適切に再生するほか、**Pitch Gate**マクロの働きにより、ピッチベンド・ホイールの操作にも反応します。

制御パネルの配置調整

インストゥルメントの制御パネルを見ると、数多くのつまみ類が、対応するマクロやモジュールごとにグループに分かれて並んでいます。したがって、例えばフィルター関連のつまみはどこにあるか、区別しやすいになっています。

この配置を調整して使いやすくしましょう。現時点では、例えば「Oscillator1」のグループと「Oscillator2」のグループが重なり合っ

いて、操作しにくい状態です。**Lock/Unlock Panel** ボタン (「スパナ」ボタン) を押して、制御パネルのロックを解除してください。ボタンが点灯し、制御パネルに格子が現れます。なお、この状態でもインストゥルメントの演奏は可能ですが、つまみ類の調節はできません。5つのグループ・パネルをドラッグして動かし、使いやすいように配置し直してください。



使いやすい配置し直した制御パネルの例

配置が決まったら、**Lock/Unlock Panel** ボタンをもう一度押して、誤って操作しても配置が変わってしまわないようにしておきます。

保存

ここで作ったシンセサイザーを保存しておけば、別のアンサンプルに組み込んで再利用することもできます。しかしその前に、「Instrument」という漠然とした名前になっているので、分かりやすい名前に変更しておきましょう。パネル・ヘッダーに「Instrument」と表示されている箇所をダブル・クリックすると、プロパティー画面が開きます。その**LABEL** 欄に、例えば「My DIY Synth」と入力し、**return** キーを押すと名前が変わります。ヘッダーにも新しい名前が表示されているので、その箇所でコンテキスト・メニューを開き、**Save Instrument as...** コマンドを実行してください。

すると **Save Instrument** ダイアログ・ボックスが現れます。保存先フォルダーとファイル名を指定し、**Save** ボタンを押してください。ファイル名は初めからインストゥルメント名と同じ「My DIY Synth」になっているので、そのままでも保存してもよいでしょう。アンサンプル全体を保存するかどうか訊ねられますが、「My DIY Synth」 インストゥルメント以外は保存しておかなくても困らないので、**No** と応えても構いません。

注意：アンサンブル全体を保存したい場合は、主メニューから **File > Save Ensemble...** コマンドを実行してください。

拡張

「My DIY Synth」を継続して使っていると、新たな機能が欲しくなるかも知れません。自由にアンサンブルの構成を変えて試せることが REAKTOR の身上です。「My DIY Synth」中のマクロを入れ替えるだけでも、さまざまな音作りの可能性が広がります。例えばエンベロープ生成器を ADR 型から ADSR 型に変えるだけでも、打撃音の響きがだいぶ変わります。

9.3. 独自のストラクチャーの構築

前節ではできあいのマクロを使ってシンセサイザーを組み立てましたが、今度は何も無いところから構築してみましょう。ひとまとまりの機能ごとにマクロとしてまとめる方法を前節では推奨しましたが、今回は階層構造を使わず、もっと基本的な部品である「モジュール」を、独自のストラクチャー・ウィンドウ上に配置していきます。というのも、ここで作るのはきわめて単純な構成なので、階層に分けて管理するとかえって分かりにくくなるからです。

基本ストラクチャーの構築

File > New Ensemble コマンドで、新しいアンサンブルを開いてください。アンサンブルのストラクチャー・ウィンドウには初めからインストゥルメントがいくつか並んでいますが、**Master**、**Audio In**、**Audio Out** を除き、すべて削除してしまいます。

ストラクチャー・ウィンドウの背景部分でコンテキスト・メニューを開き、**Insert instrument > New-2In2Out** を実行してください。「Instrument」という名前の、空のインストゥルメントが現れます。この名前を例えば「My DIY Struct」と変更してください。次に、ダブル・クリックしてストラクチャー編集画面に切り替え、不要なモジュールを削除します。**L** という出力端子と、その前段にある **Audio Voice Combiner** モジュール(名前として「J」と表示されているもの)を除いて、すべて削除してください。再び背景部分をダブル・クリックして元の表

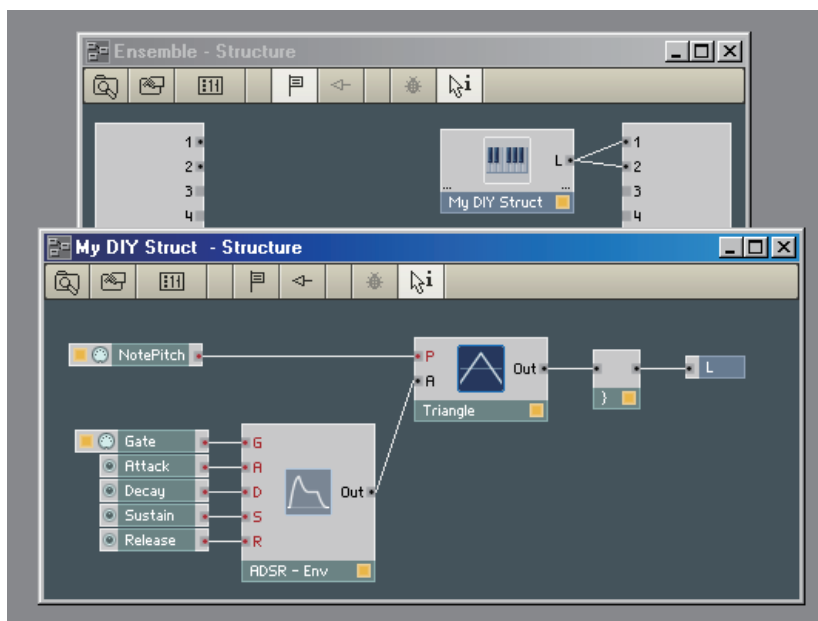
示に戻し、「My DIY Struct」の **L** 出力と、**Audio Out** の 1, 2 番入力ポートを接続してください。

再び「My DIY Struct」のストラクチャー編集画面に切り替え、ここにシンセサイザーの実体を組み込んでいくことにします。

最初に発振器を組み込みましょう。今回は三角波を生成することにするので、背景部分でコンテキスト・メニューを開き、**Built-In Module > Oscillator > Triangle** を実行してください。

次に、入力された MIDI ノート・メッセージの音量と音高を調べるモジュールが必要です。同様にコンテキスト・メニューを使って、**Built-In Module > MIDI In > Gate** および **Built-In Module > MIDI In > Note Pitch** を実行してください。さらに、エンベロープ制御のため、**Built-In Module > LFO, Envelope > ADSR** を実行して **ADSR-ENV** モジュールを追加します。

以上のモジュールを次の図のように結線します。**ADSR-Env** モジュールの **A**、**D**、**S**、**R** 入力に対する結線は、コンテキスト・メニューの **Create Control** コマンドを使うと手間がかかりません。制御つまみ類の出力を直接入力ポートにつなぐ場合は、いつでもこの方法が使えます(必要ならばいったん接続しておいてから修正)。



動作原理

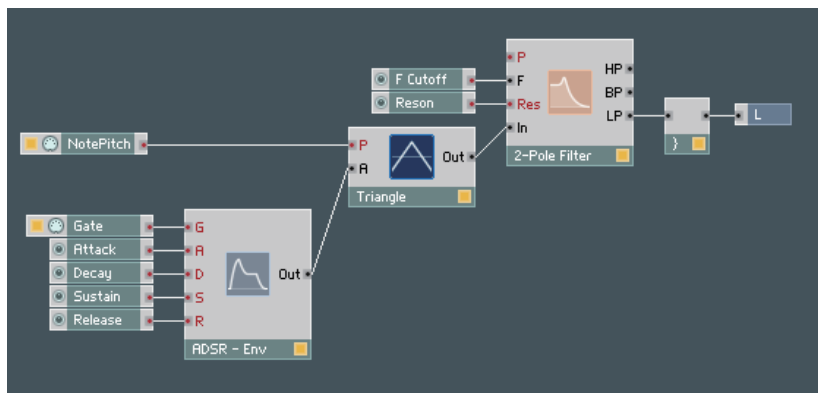
ADSR-Env モジュールは、制御パネルにある **Attack**、**Decay**、**Sustain**、**Release** の各ノブの設定に従い、信号レベルの変化を制御するためのエンベロープを生成します。**ADSR-Env** モジュールの **G** 入力にゲート信号を与えるとトリガーがかかり、エンベロープ信号の生成が始まりますが、このゲート信号は MIDI キーボードを叩くことにより発生します。どのキーを叩いたか (MIDI ノート番号) は、**Note Pitch** モジュールが認識し、この情報を発振器の **P** 入力に供給します。

これで演奏は一応できるようになりましたが、響きが単調で面白くないかも知れません。そこでフィルターの登場です。

レゾナンス・フィルターの追加

ここでは 2 ポールのフィルターで、カットオフ周波数に変調可能なものを使うことにしましょう。やはりコンテキスト・メニューを開いて、**Built-In Module > Filter > Multi 2-Pole FM** を実行します。**Triangle** モジュールの **Out** 出力と **Multi 2-Pole FM** モジュールの **In** 入力をつなぎ、さらに、**Multi 2-Pole FM** モジュールの **LP** 出力と **Audio Voice Combiner** の入力をつないでください。

次に、先ほど説明したのと同じ **Create Control** コマンドを使って、フィルターの **F** 入力、**Res** 入力に、それぞれカットオフ周波数、レゾナンスを調整するつまみ類を接続します。



フィルターの機能

F Cutoff ノブ、**Reson** ノブをいろいろ調節しながら、MIDI キーボードを演奏してみてください。なお、配置が不適切で操作しにくい場合は、いったんロックを解除して配置を調整し、再びロックしてください。**Lock/Unlock Panel** ボタンが隠れている場合は、コンテキストメニューからもロック / ロック解除の切り替えが可能です。



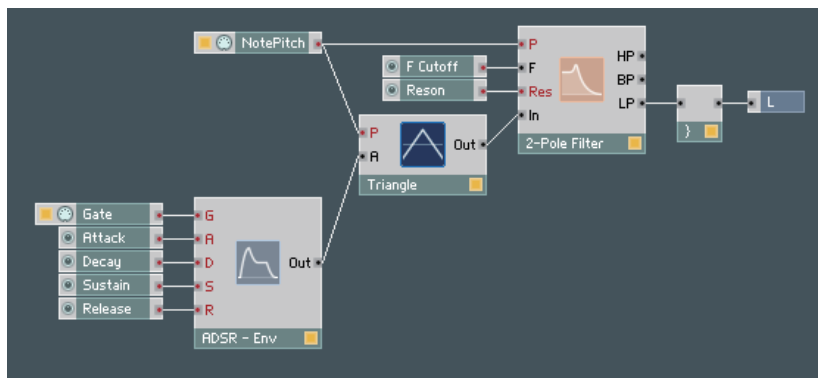
F Cutoff はカットオフ周波数を表し、フィルターの **LP** 出力からは、この周波数以上の帯域を減衰させた信号が現れます。**Multi 2-Pole FM** モジュールからはこのほかに、ある周波数帯域の上下を減衰させた信号 (**BP**)、カットオフ周波数以下の帯域を減衰させた信号 (**HP**) も得られます。

Reson はレゾナンス、すなわち、カットオフ周波数付近の帯域を強調する度合いを表します。このノブを右いっぱい回して 9.5 程度以上にすると、自己発振を起こすようになります。非常に大きな音が出てスピーカーや耳に悪影響を及ぼす恐れがあるので注意してください。

カットオフ周波数を音域に追従させる機能

低音と高音を弾き比べてみると、高音が少々聴こえにくいかも知れません。これはカットオフ周波数が音域にかかわらず固定であるためです。カットオフ周波数以上の音を演奏すると、信号全体が減衰してしまうのです。これを回避するためには、カットオフ周波数を音域に追従して変化させる必要があります。**Note Pitch** モジュールの出力を、カットオフ周波数を調整する **P** 入力につないでください。

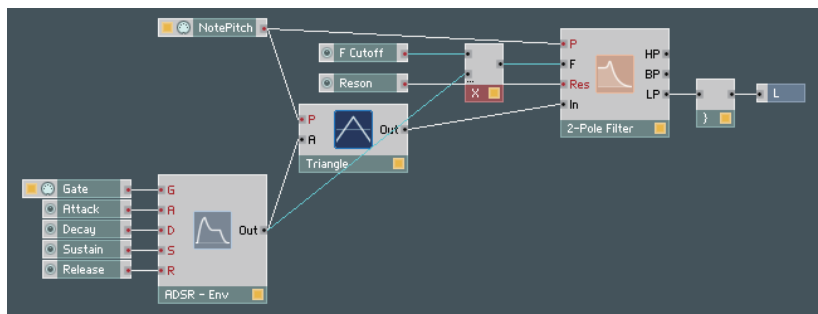
このフィルター・モジュールにはカットオフ周波数の制御入力がある **P** と **F** の 2 つあり、加算して最終的なカットオフ周波数を決めるようになっています。そのため、高音域でも信号全体が減衰してしまうことなく、想定通りの音が聴こえます。



フィルター・エンベロープの追加

最後に、エンベロープに沿ってカットオフ周波数を制御してみましょう。ここでは単純にするため、既に使っている **ADSR-Env** モジュールの出力を流用します。もっと細かく制御したい場合は、フィルター専用のエンベロープ生成器を追加してください。

この処理のためには乗算器が必要です。**Built-In Module > Math > Multiply** で追加し、**ADSR-Env** の出力を一方の入力、**F Cutoff** の出力 (ノブの設定値) をもう一方の入力につないでください。この 2 つを乗算した結果を、**Multi 2-Pole FM** モジュールの **F** 入力に与えます。



実際に何か演奏してみると、フィルター・エンベロープの効果が分かるでしょう。**ADSR-Env** の出力値は 0～1 の間で変化するので、これに応じてノブで設定したカットオフ周波数が変化します。リリース段に入るとカットオフ周波数も下がっていくので、フィルターを通過する帯域が下の方に狭まり、徐々に音が消えていきます。

この効果の度合いを「エンベロープ変調強度」と言います。これを反映するよう、ノブの名前も「F Cutoff to Env Mod」などとするでしょう。モジュールをダブル・クリックし、ラベルを変更してください。

その他の拡張

そのほか、必要に応じ、次のような機能拡張も試してみるとよいでしょう。

- ・ フィルターの **HP**、**BP** 出力の活用。
- ・ 2 ポールから 4 ポールのフィルターに変更。
- ・ 他のエンベロープ生成器に置き換え。
- ・ フィルター専用のエンベロープ生成器を追加。

ほかにもさまざまな拡張が考えられます。ハードウェアと違って、いろいろ試してみても REAKTOR を壊してしまう心配はいりません。

但し 1 点だけ注意しておきましょう。試聴中に突然大きな音が出て、スピーカーや耳に悪影響を及ぼす恐れがあります。アンプの音量は絞り気味にして実験してください。

10. REAKTOR の基本操作

REAKTOR のユーザー・インターフェイスは、基本的に Windows や MacOS X の一般的な使い方に倣っているので、多少でも使ったことがあればすぐに慣れるでしょう。ここでは REAKTOR に特有の操作を中心に、基本操作について簡単に説明しておきます。

10.1. マウス

REAKTOR の機能は、マウスだけでもほとんどが実行可能です。

- ・ **操作対象の選択。** マウス・カーソルを操作対象の上に持っていき、左ボタンを押して (クリックして) ください。するとラベルが赤に変わるので、選択状態であることが分かります。複数の操作対象をまとめて選択したい場合は、Windows ならば **Ctrl** キー、MacOS X ならば **shift** キーを押しながら、順次左ボタンを押してください。また、背景部分で左ボタンを押し、そのままドラッグして放すと、この範囲内にある操作対象をまとめて選択できます。
- ・ **操作対象の位置移動。** マウス・カーソルを操作対象の上に持っていき、左ボタンを押して (クリックして) そのままドラッグしてください。複数まとめて移動したい場合は、該当するものをすべて選択状態にしておいてから、いずれか 1 つを、同様に操作して移動します。それぞれが結線で結ばれていれば、その状態を保ったまま、位置だけが変わります。なお、格子を設定しておけば、自動的にその格子位置に揃います。
- ・ **結線による接続。** モジュールやマクロの出力ポート上にマウス・カーソルを持っていき、左ボタンを押して (クリックして) そのままドラッグしてください。接続先の入力ポート上で放すと、結線で接続されます。これは信号が伝わる経路を表します。なお、逆向き、すなわち入力ポートから出力ポートに向かって接続しても、結果は同じになります。
- ・ **ダブル・クリック。** 左ボタンをダブル・クリックしたときの動作は、マウス・カーソルが何の上にあったかによって異なります。(次に説明する) コンテキスト・メニューを開いたときに、太字で表示されているコマンドを実行するのと同じ効果があります。
- ・ **コンテキスト・メニュー。** Windows ならば右ボタンを押す、MacOS X ならば **control** キーを押しながらマウス・ボタンを押す、という操

作で、コンテキスト・メニューが現れます。ここに並んでいるメニュー項目は、マウス・カーソルが何の上にあったかによって異なります。これについては節を改めて説明します。

10.2. コンテキスト・メニュー

コンテキスト・メニューは、マウス・カーソルの位置にある「操作対象」に応じてメニュー項目が異なりますが、いずれにしてもモジュール、マクロなど操作対象に対して適用できるコマンドが並んでいます。コマンドを実行すればコンテキスト・メニュー自体は画面から消えます。例えば **Delete** コマンドがコンテキスト・メニュー上にあれば、その操作対象、例えばモジュールを削除することができます。

10.3. キーコマンド

コマンドの多くは、コンピュータ側キーボードのあるキー（または複数のキーの組み合わせ）を押して起動することができます。具体的なキーは、該当するコマンド・メニューに添えて表示されています。

10.4. アンサンブル制御パネル、ストラクチャー・ウィンドウ

REAKTOR では主に、アンサンブル制御パネル・ウィンドウ (**Ensemble Panel**) およびストラクチャー・ウィンドウ (**Structure**) を使って作業します。アンサンブル制御パネルには、アンサンブル全体を制御するつまみ類と、これを構成する各インストゥルメントを制御するつまみ類が、それぞれグループ分けした状態で並んでいます。一方ストラクチャー・ウィンドウは、アンサンブル、インストゥルメント、基本マクロ、コア・セル、コア・マクロなど（まとめて「オブジェクト」と呼びます）の内部構成を表示、編集するために使います。

アンサンブル制御パネル・ウィンドウは常に 1 つだけ表示されます。ストラクチャー・ウィンドウも通常は 1 つですが、必要であればいくつでも開けます。**Alt** キーを押しながらオブジェクト上をクリックするか、コンテキスト・メニューから **Structure Window** を実行してください。もっとも、むやみに開くと画面が錯綜するので、必要最小限にとどめるようお勧めします。

ウィンドウ自体の操作方法を以下に示します。

- ・ **アンサンブル制御パネル・ウィンドウを開く** : **View > Show Panel** コマンドを実行してください。また、ストラクチャー・ウィンドウの背景部分をダブル・クリックすると順次上位階層の表示に切り替わり

ますが、最上位まで到達すると次はアンサンプル制御パネル・ウィンドウになります。

- ・ **ストラクチャー・ウィンドウを開く**：内部構成を表示したいオブジェクト上をダブル・クリックしてください。ストラクチャー・ウィンドウの内容が切り替わります。但し **Alt** キーを押しながら同様に操作すると、新しいウィンドウが開きます。
- ・ **ウィンドウの切り替え**：開いているウィンドウの名前は **View** メニューの一番下に並んでおり、ここで切り替える（最前面に出す）ことができます。画面上に見えていれば、そのウィンドウをクリックするだけでも同じ効果があります。
- ・ **上位階層への移動**：ストラクチャー・ウィンドウの背景部分をダブル・クリックすると順次上位階層の表示に切り替わります。
- ・ ウィンドウの移動、大きさ変更、最小化、および閉じる操作については、Windows/MacOS X の一般的なアプリケーションの操作方法と同じです。中身がウィンドウに納まらない場合は右と下にスクロール・バーが現れます。

Windows 版の場合は次のような動作になっています。

- ・ ウィンドウはすべて、REAKTOR 全体のアプリケーション・ウィンドウ（「REAKTOR ウィンドウ」）内に包含されています（マルチ・ウィンドウ・モード）。REAKTOR ウィンドウを大きさ変更/最小化すると、その中のウィンドウも影響されますし、他のアプリケーションが前面に來たときも同様です。
- ・ 包含されているいずれかのウィンドウを最大化すると、（画面全体ではなく）REAKTOR ウィンドウ全体に広がります。1 つのウィンドウに対して最大化の操作を施すだけで、全ウィンドウが最大化されます。逆にいずれかのウィンドウの最大化を解除すると、他のウィンドウも一斉に解除されます。
- ・ 最小化したウィンドウは、タイトル部分だけが主 REAKTOR ウィンドウの下部に現れます。
- ・ **Ctrl+Tab** キーで、各ウィンドウを順次前面に出すことができます。

11. メニュー・コマンド

コンテキスト・メニューとは別に、アプリケーション全体に関わる機能コマンドが並んだメニューがあります。この章では各コマンドについて解説します。

11.1. File メニュー

New Ensemble

新規アンサンブルを作成します。**Master** インストゥルメントと、**Audio In** および **Audio Out** の 2 つのモジュールが、最初から追加された状態になっています。

Open...

アンサンブル・ファイル (*.ens) をハード・ディスクから読み込みます。

Save Ensemble

アンサンブルをファイル (*.ens) に保存します。これに含まれるインストゥルメント、ストラクチャー、制御パネル、スナップショットも組にして保存します。

Save Ensemble As...

Save Ensemble と同じ機能ですが、保存先フォルターとファイル名を指定できます。

Save Window As...

選択状態のウィンドウの中身を、ファイル名をつけて保存します。

アンサンブル制御パネル・ウィンドウを選択している場合は、実質的に **Save Ensemble As...** コマンドと同じ機能です。

インストゥルメントのストラクチャー・ウィンドウを選択している場合は、対応するインストゥルメントを *.ism ファイルに保存します。これに含まれるストラクチャー、制御パネル、スナップショットも組にして保存します。

マクロのストラクチャー・ウィンドウを選択している場合は、マクロ全体を、そのストラクチャーと組にして ***.mdl** ファイルに保存します。

Import MIDI File...

REAKTOR には MIDI ファイル・プレイヤーが統合されており、SMF(Standard MIDI File) 形式のファイルを読み込んで再生できます。MIDI ファイルはほとんどのシーケンサーでも作成できます。Windows の場合、MIDI ファイルには拡張子 ***.mid** がついています。

組み込みの MIDI ファイル・プレイヤーを使えば、別にシーケンサーを用意しなくても再生できます。REAKTOR を生演奏に使う場合には特に有用な機能でしょう。仮にこれがなかったとすると、同じコンピューター上で、並行してシーケンサーも動かす必要が出てきます。するとさまざまな問題が生じやすいばかりでなく、同じファイルをシーケンサーと REAKTOR の両方に読み込ませなければならないなど、操作も複雑になりがちです。何よりも、2つのプログラムを切り替えながら、さまざまなパラメーターを調整しなければなりません。

さらに、サンプル単位の正確なタイミング調整が可能、という長所があります。MIDI ファイル上で同時に鳴るよう設定された音は、REAKTOR で再生すると高い精度で同時に鳴り始めるのです。もちろんその分解能や精度は、MIDI ファイルを生成したシーケンサーの能力にも依存します。

MIDI ファイルの読み込みは、手動と自動のどちらでも可能です。このコマンドを起動したときに開くファイル選択ダイアログ・ボックスでは、標準 MIDI ファイルも選択できるようになっています。一方、アンサンプルを開くとき、それと拡張子を除いて同じ名前の MIDI ファイルが同一フォルダー内にあれば、自動的に読み込まれます。

Settings メニューには、MIDI ファイル・プレイヤーに関する設定項目が3つあります。**Play MIDI File** をオンにすると、REAKTOR のクロックを起動すれば MIDI ファイルの再生が始まるようになります。**Loop MIDI File** がオンならば、同じ MIDI ファイルが繰り返し再生されるようになるので、短い楽句を繰り返し演奏したい場合などに有用でしょう。最後に、**Ignore Tempo Change** をオンにすると、MIDI ファイル中にテンポ・チェンジ・イベントがあっても無視し、REAKTOR のクロックに合わせて再生するようになります。

MIDI ファイル・プレイヤーの操作機能は、REAKTOR のクロック操作ボタンで制御できます。

- ・ **Start/Restart Clock** ボタンを押すと MIDI ファイルの先頭から再生が始まります。
- ・ **Pause/Stop Clock** ボタンを 1 回押すと一時停止になります。もう一度押すと停止になり、再生位置は MIDI ファイルの先頭に戻ります。

Batch Processing

REAKTOR 3 で作成したファイルを REAKTOR 5 の形式に変換する、あるいはオーディオ・ファイルを分析してグラニュラー・サンプラー・モジュールで扱えるようにするコマンドです。REAKTOR 3 USB キーを用意し、変換元と変換先のフォルダーを指定して **OK** を押してください。

(最近使用したアンサンプル)

最近使用したアンサンプル名が 8 回分メニューに列挙されているので、迅速に再読み込みできます。

Exit

開いているウィンドウをすべて閉じ、REAKTOR プログラムを終了します。必要に応じ、ファイルに保存してから終了するかどうか訊ねられます。

11.2. Edit メニュー

Undo

直近の編集操作を取り消して元に戻します。但し制御パネル上のつまみ類の調整に関しては取り消すことができません。必要ならば **Snapshots** ウィンドウの **Compare** ボタンを使ってください。

取り消しできる最大回数は調整可能です。**Preference** コマンドの、**Options** ページに関する説明を参照してください。メモリーが不足気味の場合は、少なめに設定するとよいでしょう。

Redo

Undo コマンドで取り消した操作を再実行します。**Undo** を実行した回数だけ再実行できます。

Cut

選択状態のオブジェクトをクリップボードに複製し、編集画面上からは削除します。

Copy

選択状態のオブジェクトをクリップボードに複製します。

Paste

クリップボード上のオブジェクトを、編集画面上の指定した位置に複製します。

あらかじめ、ストラクチャー・ウィンドウ上で、貼り付ける位置をクリックしておいてください。

Duplicate

選択状態のオブジェクトを複製します。**Copy** と **Paste** を続けて実行するのと同等了。

Delete

選択状態のオブジェクトを削除します。

Select All

最前面のウィンドウにあるオブジェクトをすべて選択状態にします。少数の例外を除く大部分のオブジェクトを選択したい場合も、このコマンドですべて選択しておき、不要なオブジェクトの選択を解除すると手間が省けます。Windows ならば **Ctrl** キー、MacOS X ならば **shift** キーを押しながら、不要なオブジェクトをクリックしてください。

11.3. Settings メニュー

Sample Rate

REAKTOR で生成、処理するオーディオ信号のサンプル・レートを設定します。値を大きくすれば音質が上がりますが、CPU に対する負荷も高くなります。内部処理用のサンプル・レートはサブ・メニューから自由に選択できますが、選択肢はサウンド・カードやホスト・プラグ・インによって異なります。REAKTOR 側の設定が、サウンド・カードやホスト・プラグ・インの設定と一致していなければ、**Audio In**、**Audio Out** モジュールで変換されるようになります。

Control Rate

1 秒間に何回の割合でイベント信号の値を更新するか、その頻度を設定します。**LFO**、**Slow Random**、**Hold**、**A to E**、**Event Smoother** など、多くの基本モジュールのイベント処理に影響します。サンプル・レートに比べると値がずっと小さいので、CPU に対する負荷もそれほど高くありません。したがって、オーディオ信号、イベント信号のどちらを使っても結果に違いがない場合は、イベント信号で制御するようにしてください。

頻度を高くすると、より細かな分解能で信号を処理できるようになります。

MIDI Learn

制御パネル上のつまみ類を MIDI Learn モード、すなわち外部の MIDI コントローラーとの対応を設定するモードにします。この状態で MIDI コントローラーを操作するだけで対応がつき、通常モードに戻ります。なお、アンサンプル制御パネルのツールバーにも、同じ機能のボタン (MIDI コネクターをかたどったアイコン) があります。

Set Protected / Set Unprotected

誤操作によりつまみ類の位置を動かしてしまわないよう保護するモードのオン / オフを切り替えます。ツールバーにも同じ機能のボタンがあります。

Automatic Layout

インストゥルメントの制御パネルをすべて、配置を自動調整するモードにします。なお、アンサンブルのプロパティ画面の **Appearance** ページにある、**Automatic Panel Layout** でもモードを切り替えることができます。初期状態ではオンになっています。

External Sync

Sync Clock、**1/96 Clock** モジュールで、REAKTOR 内蔵クロックを使うか、外部クロック源を MIDI 経由で受け取って使うか、を切り替えます。また、**Start/Stop** モジュールを、外部からの MIDI-Start/Stop メッセージで制御するかどうか、切り替える働きもあります。オンにすると、主ツールバーの **BPM** 欄の値は無視し、外部クロックに同期して動作するようになります。

MIDI Clock Out

オンにすると、MIDI 出力ポートに MIDI クロックを送信するようになります。送信先 MIDI 出力ポートは、**Audio Setup** 画面の **MIDI** ページで選択します。

Clock Start

REAKTOR のマスター・クロックを起動します。クロックに同期して動作するモジュールが稼動し始めます。また、**Start/Stop** モジュールが「開始」イベントを生成します。なお、アンサンブル制御パネルのツールバーにある **Start/Restart Clock** ボタンも同じ機能です。

Clock Stop

REAKTOR のマスター・クロックを停止します。**Sync Clock**、**1/96 Clock** モジュールも稼動を停止します。また、**Start/Stop** モジュールが「停止」イベントを生成します。なお、アンサンブル制御パネルのツールバーにある **Pause/Stop Clock** ボタンも同じ機能です。

Play MIDI File、Loop MIDI File、Ignore Tempo Change

REAKTOR 内蔵の MIDI ファイル・プレイヤーに関する設定です。詳しくは **Import MIDI File...** コマンドの説明を参照してください。

11.4. System メニュー

Run/Stop Audio

REAKTOR のオーディオ処理を開始 / 停止します。REAKTOR 全体のオン / オフに相当します。なお、主ツールバーの **Run/Stop Audio** ボタンも同じ機能です。

Debug

こ こ に は **Measure CPU Usage**、**Show Module Sorting**、**Show Event Initialization Order** の各コマンドが並んでいます。

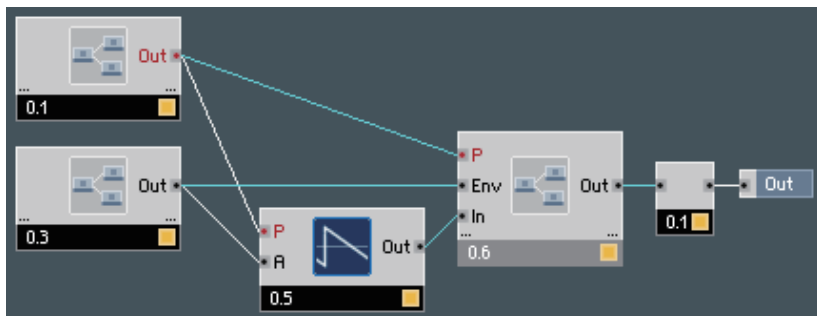
Measure CPU Usage

オーディオ処理に関わる、インストゥルメント、マクロ、モジュールがすべて、CPU に対する負荷を計測するモードになります。ストラクチャー表示に切り替えると、名前が表示されていた箇所 (背景が黒の帯) が、負荷の表示になります。負荷が高いとき、どこで CPU の処理能力が大きく消費されているかを判断する手掛かりになるでしょう。これをもとに構成を見直し、余裕を持って処理できるよう改善してください。

但し、オーディオ処理がオフになっている、あるいはもっぱらイベント処理のみ行うコンポーネントの場合は、負荷が表示されません。

なお、計測処理自体も負荷となるため、通常動作モード時の負荷とは若干の違いが生じます。

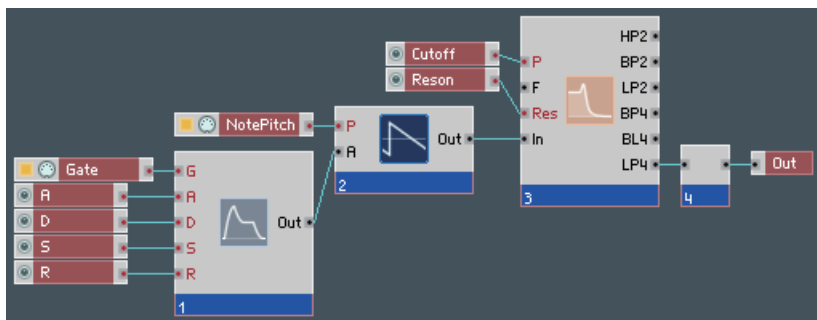
このモードに切り替えるためには、あらかじめオーディオ処理を開始しておく必要があります。また、計測中はオーディオ出力がありません。



CPU に対する負荷を表示している様子

Show Module Sorting

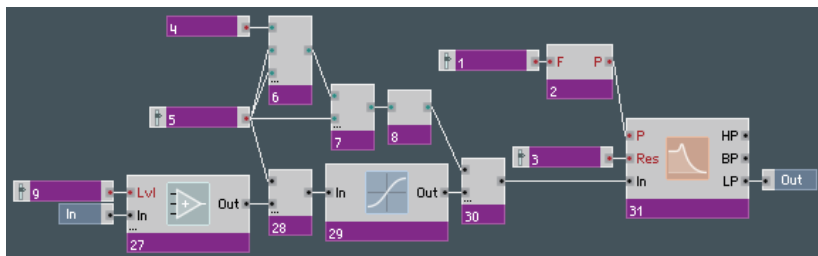
オーディオ・モジュールの処理順序番号を表示するモードに切り替えます。オーディオ信号を処理する各モジュールについて処理の順序を計算し、青い帯上に表示します。



モジュールの処理順序番号を表示している様子

Show Event Initialization Order

初期設定イベントがあったときに各モジュールが初期化される順序を表示します。



初期設定の順序番号を表示している様子

Audio + MIDI Settings...

オーディオ /MIDI インターフェイスの設定ダイアログ・ボックスを開きます。詳しくは第 6 章「スタンドアローン版の REAKTOR 5」を参照してください。

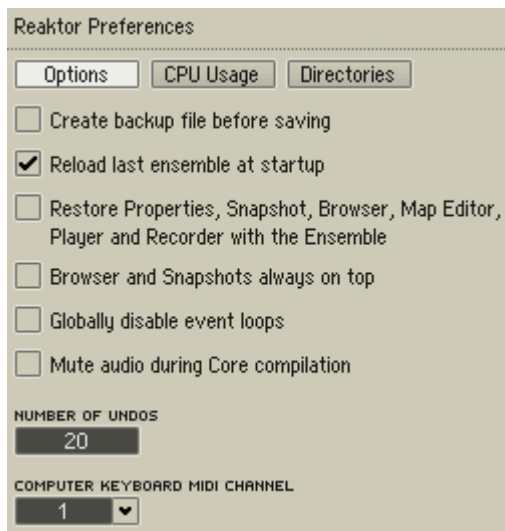
OSC Settings...

OSC Setup ウィンドウを開きます。詳しくは第 8 章「OSC(Open Sound Control)」を参照してください。

Preferences

REAKTOR 全体の環境設定画面を開きます。これはさらに、**Options**、**CPU Usage**、**Directories** の各ページに分かれています。

Options ページ



- **Create backup file before saving:** ファイルに保存する際、同じ名前のファイルが既にあった場合は、拡張子を ***.bak** にして元のファイルを残すようになります。操作を誤った場合でも、拡張子を元に戻せば復元できます。
- **Reload last ensemble at startup:** REAKTOR を起動すると、前回開いていたアンサンブルが自動的に開くようになります。
- **Restore Properties, Snapshot, Browser, Map Editor, Player and Recorder with the Ensemble:** ウィンドウやダイアログ・ボックスの位置や大きさが、アンサンブルを前回保存したときと同じ状態になります。
- **Browser and Snapshots always on top:** ブラウザーやスナップショット・ウィンドウが、他のダイアログ・ボックスやウィンドウよりも常に前面に表示されるようになります。
- **Globally disable event loops:** オンにすると、イベント信号のループが生じそうになったとき、ループの発生源を表示して続行するかどうか訊ねるようになります。ループが生じると、処理スタックがあふれ、再生不能、あるいは操作不能に陥る可能性があります。その場合、いったん REAKTOR を終了して再起動し、**Globally disable event**

loops をオンにして動かして、表示されるループ発生源を手掛かりに問題点を特定してください。このときオーディオ出力もオフにすると、ループが何重にも重なって制御しきれなくなるのをある程度防ぐことができます。REAKTOR の動作を安定させるためにも、通常はオンにしておくようお勧めします。なお、旧版で保存したファイルを読み込んだ場合は、互換性を保つためオフになっています。

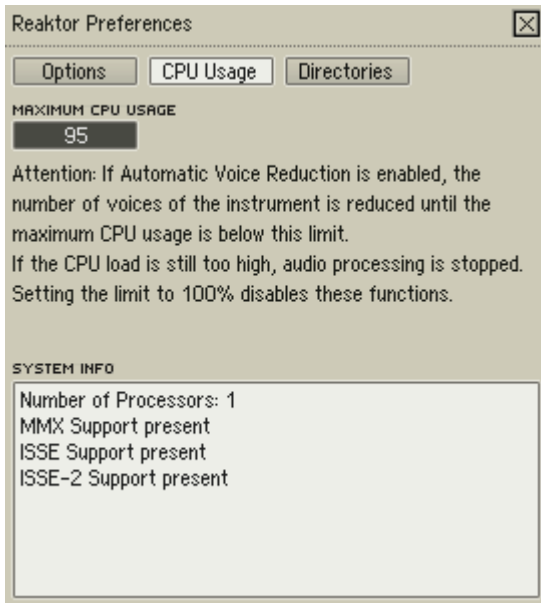
注意：多くの場合、**Iteration** モジュールを使うことにより、イベント・ループを使わなくても同等の処理ができます。このモジュールには処理速度を制限するプロパティがあるので、ループが速過ぎる場合に生じる雑音も回避できます。

- ・ **Mute audio during Core compilation:** コア・オブジェクトのコンパイル中はオーディオ出力を停止します。コンパイル処理がその分高速になります。
- ・ **NUMBER OF UNDOS:** **Undo** コマンドで操作を取り消せる回数を指定します。例えば 20 と設定すれば、20 回分まで遡って元の状態に戻せることになります。なお、0 とすれば **Undo** コマンド自体が使えなくなります。

アンサンブル中に大容量のオーディオ・ファイルが含まれる場合、元の状態を記憶しておくために大量のメモリーが必要になります。メモリー容量に制約がある場合は小さめの値を設定してください。

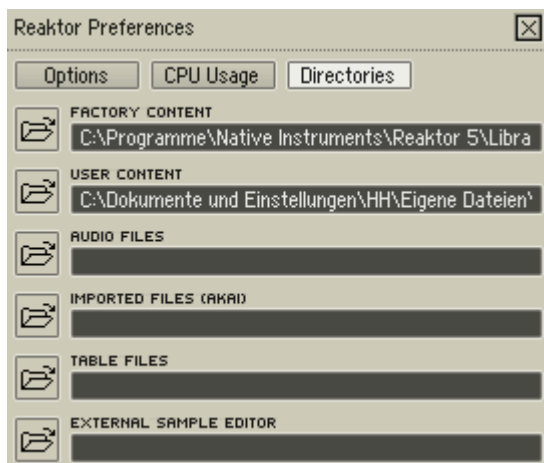
- ・ **COMPUTER KEYBOARD MIDI CHANNEL:** MIDI キーボードの代用としてコンピューター側のキーボードを使う場合の、MIDI チャンネル番号を設定します。初期状態では 1 になっています。これは、インストゥルメントが受け付ける MIDI チャンネル番号が、初期状態では 1 になっているのに合わせたものです。インストゥルメントの方で (例えばチャンネル番号 2 というように) 設定を変えている場合は、それに合わせて変更してください。

CPU Usage ページ



- ・ **MAXIMUM CPU USAGE:** CPU に対する負荷がある限度に達すると、多声インストゥルメントの同時発音声部数を、処理能力に応じて自動的に減らすことができます。この限度を % 単位で設定してください。なお、この機能により声部数が減るのは、インストゥルメントのうち、**Automatic Voice Reduction** プロパティ (Function ページで設定) がオンになっているものに限りです。また、「100%」と設定すると、声部数の自動削減処理が実質的に無効になります。

Directories ページ



- ・ **FACTORY CONTENT:** REAKTOR に付属するアンサンブル、インストゥルメント、基本マクロ、コア・セル、コア・マクロの格納場所を表します。インストール処理の際、自動的にこの位置にコピーされます。ここにあるオブジェクトは、コンテキスト・メニューを開いたり、ブラウザー上のボタン (**ENS.**、**INSTR.**、**MACRO**、**CORE C.**、**CORE M.**) を押したりしたときに列挙されるので、呼び出す際も手間がかかりません。
- ・ **USER CONTENT:** 独自に作成したオブジェクトの格納場所を表します。アンサンブル、インストゥルメント、基本マクロ、コア・セル、コア・マクロ、オーディオ・ファイル、インポートしたファイル、画像ファイル、スナップショット、テーブルなどがこれに該当します。**FACTORY CONTENT** と同様に、コンテキスト・メニューやブラウザー・ボタンを使って簡単に呼び出せます。

警告：独自に作成したファイルを、**FACTORY CONTENT** で示される場所には置かないでください。REAKTOR の更新時に削除されてしまう恐れがあります。

- ・ **AUDIO FILES:** オーディオ・ファイル (*.wav、*.aif、*.aiff) の格納場所を表します。プレイヤー、レコーダー、サンプル・マップ・エディ

ター、テープデッキ、サンプラー・モジュールなどがこの設定を参照します。

- ・ **IMPORTED FILES (AKAI):** サンプル・マップ・エディターの **Akai-Import** コマンドで変換したマップ・ファイル (*.map) の格納場所を表します。
- ・ **TABLE FILES:** テーブル・ファイル (*.ntf) の格納場所を表します。**Audio Table**、**Event Table** モジュールのプロパティ画面からファイルを読み書きする際、この設定を参照します。
- ・ **EXTERNAL SAMPLE EDITOR:** サンプル・マップ・エディターの **Edit Sample List** メニューから、**Edit** コマンドを実行したときに起動されるエディターを設定します。

11.5. View メニュー

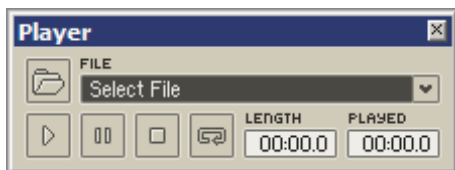
Show/Hide Hints

ボタン、入出力ポートなどの上にマウス・カーソルを持っていったときに、その説明を表示するか否かを切り替えます。

Show/Hide Toolbox

主ツールバーの表示 / 非表示を切り替えます。詳しくは第 12 章「REAKTOR のツールバー」を参照してください。

Show/Hide Playerbox





内蔵プレイヤーの表示 / 非表示を切り替えます。

オーディオ・ファイル (*.wav、*.aif、*.aiff) を読み込み、フィルター、ディレイその他の加工を施して再生することができます。アンサンプル・ストラクチャー・ウィンドウで見ると、**Audio In** モジュールの上 2 つの入力ポートにオーディオ信号が供給されているのと同じ状態です。

重要: **Audio In** モジュールの上 2 つの入力ポートにオーディオ信号を送らなければ、音となって出力されません。


プレイヤーには次のような操作ボタン類がついています。


- ・  **Load:** ハード・ディスクからオーディオ・ファイルを読み込み、再生を始めます。

- ・  **FILE:** オーディオ・ファイルを読み込むと、同じフォルダーにある他のオーディオ・ファイル名がこのドロップ・ダウン・メニューに列挙されます。追加で読み込む場合に操作が簡単になります。

- ・  **Play:** 読み込んだオーディオ・ファイルを再生します。


注意: サンプル・レートは、主ツールバーの **Sample Rate** 設定に従います。

- ・  **Pause:** 再生を一時停止します。もう一度押すと続きの再生が始まります。

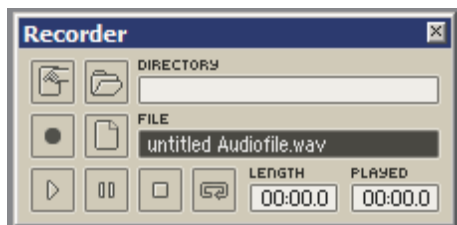
- ・  **Stop:** 再生を停止し、先頭位置に戻ります。

- ・  **Loop:** 繰り返し再生モードにします。

- ・  **LENGTH** 表示: オーディオ・ファイルの長さを表示します。

- ・  **PLAYED** 表示: 現在の再生位置を表示します。

Show/Hide Recorderbox




内蔵レコーダーの表示 / 非表示を切り替えます。

アンサンプルの出力信号を、オーディオ・ファイル (Windows: *.wav、MacOS X: *.aif) の形で、ハード・ディスク中に直接保存します。アンサンプル・ストラクチャー・ウィンドウで見ると、**Audio Out** モジュールの上 2 つの出力ポートが録音対象になります。

レコーダーには **Audio Out** モジュールの出力を再生する機能も付属しています。


重要： **Audio Out** モジュールの上 2 つの出力ポート以外のオーディオ信号は、音となって出力されません。

レコーダーには **Recorder Settings** ダイアログ・ボックスがあり、録音開始 / 停止操作に関するさまざまな設定ができます。画面の左上にある  ボタンを押してください。











ここでは次のような設定ができます。

- **RECORD START BY:** どのようなきっかけで録音が始まるか、を設定します。**Manual: Play/Pause** ボタンを押したとき。**Note On:** MIDI Note On イベントを受け取ったとき。**Clock Start:** 主ツールバーの **Start/Restart Clock** ボタンを押したとき。
- **RECORD STOP BY:** どのようなきっかけで録音を終了するか、を設定します。**Manual: Stop** ボタンを押したとき、**Pause** ボタンを押したとき (一時停止)。**Note Off:** MIDI Note Off イベントを受け取ったとき。**Clock Stop:** 主ツールバーの **Start/Restart Clock** ボタンを押したとき。**Loop Length: LOOP LENGTH (BARS)**(後述)としてあらかじめ設定しておいたループ長に達したとき。
- **START OFFSET (BARS):** 録音開始イベントを受け取ってから実際に開始するまでの遅延を、小節単位で指定します。ある音のエコー効果部分だけを録音し、その音自体は除きたい場合などに使うとよいでしょう。
- **LOOP LENGTH (BARS):** 録音時間を小節単位で指定します。**RECORD STOP BY** として **Loop Length** をオンにした場合に有効です。
- **Slave Player Controls to Recorder:** オンにすると、プレイヤーもレコーダー側の再生 / 停止ボタンに反応して動作するようになります。
レコーダー本体には次のようなボタン類がついています。

-  **Load:** 録音先ファイルを指定します。既存のファイルと同じ名前を指定すると、上書きされてしまうので注意してください。また、既存のファイルの再生にも使います。

-  **DIRECTORY:** 録音ファイルの保存先フォルダーを表示します。

-  **Record:** 録音を開始できる状態にして待機するボタンです。このあと **Pause** ボタンを押すとすぐに録音が始まります。
-  **New:** このあとの録音に使うオーディオ・ファイルを新しく作るボタンです。

- ・ **FILE:** 録音先オーディオ・ファイル名を表示します。ファイル名を変更したい場合はここに入力してください。
- ・  **Play:** 読み込んだオーディオ・ファイル (**File** 欄に表示されているもの) を再生します。
- ・  **Pause:** 再生または録音を一時停止します。もう一度押すと続きます。
- ・  **Stop:** 再生または録音を停止し、先頭位置に戻ります。
- ・  **Loop:** 繰り返し再生モードにします。
- ・  **LENGTH** 表示: オーディオ・ファイルの長さを表示します。
- ・  **PLAYED** 表示: 現在の再生位置を表示します。

Show/Hide Properties

選択したオブジェクトに関する、プロパティ画面の表示 / 非表示を切り替えます。標題の表示部分をダブル・クリックする、コンテキスト・メニューから **XXXX Properties** コマンドを実行する、といった手順でも切り替えが可能です (「XXXX」には該当するオブジェクト名が入るので、実際には「Fader Properties」などというコマンド名)。ダイアログ・ボックス自体は常に関いたままにしておいても構いません。この場合、オブジェクトの選択に応じて中身が切り替わります。

Show/Hide Sample Map Editor

サンプル・マップ・エディターを開く / 閉じるコマンドです。

Show/Hide Browser

ブラウザー・ウィンドウを開く / 閉じるコマンドです。

Show/Hide Snapshots

スナップショット・ウィンドウを開く / 閉じるコマンドです。

Reset All Tool Window Positions

各ウィンドウの位置や大きさは記録されているので、次に開いたときも前回の状態になりますが、この記録を捨てて初期状態に戻すコマンドです。ウィンドウが画面からはみ出すなど、問題が生じたときには実行してみてください。

Show/Hide Panel

アンサンプル制御パネル・ウィンドウを開く / 閉じるコマンドです。

Store Panelset

アンサンプルごとに、最大 8 通りのパネルセットを管理できます。これは操作を効率化するための機能で、各ウィンドウを配置した状態を登録しておき、必要に応じて切り替えながら作業できます。

アンサンプル制御パネル・ウィンドウ内に各インストゥルメントを配置し、位置や大きさ、表示 / 非表示などを使いやすいように決めてください。この状態で **Store Panelset** メニュー以下の 1 ～ 8 の番号を選択すると、その番号のパネルセットとして保存されます。

また、ツールバーにある  **Panelset** バーでも同じ操作ができます。

これを使う場合、8 通りまでという制限はありません。



3 通りのパネルセットが登録された状態の **Panelset** バー

Recall Panelset

パネルセットとして登録されている画面の状態に戻します。**Recall Panelset** メニュー以下には 1 ～ 8 番の選択肢があるので、該当する番号を選択してください。

Panelset バーでも同じ操作ができます。8 通りまでという制限はありません。

Close All Structures

ストラクチャー・ウィンドウをすべて閉じます。

Cascade

開いているウィンドウをすべて、少しずつ位置をずらしながら重ねて配置します (Windows の場合のみ)。

Tile Horizontally

ウィンドウを横に並べて配置します (Windows のみ)。

Tile Vertically

ウィンドウを縦に並べて配置します (Windows のみ)。

Minimize

ウィンドウをタイトル部分のみ残して最小化します (MacOS X のみ)。

Close

ウィンドウを閉じます (MacOS X のみ)。

Arrange Icons

最小化されたウィンドウの配置を整頓します (Windows のみ)。

(開いているウィンドウ名のリスト)

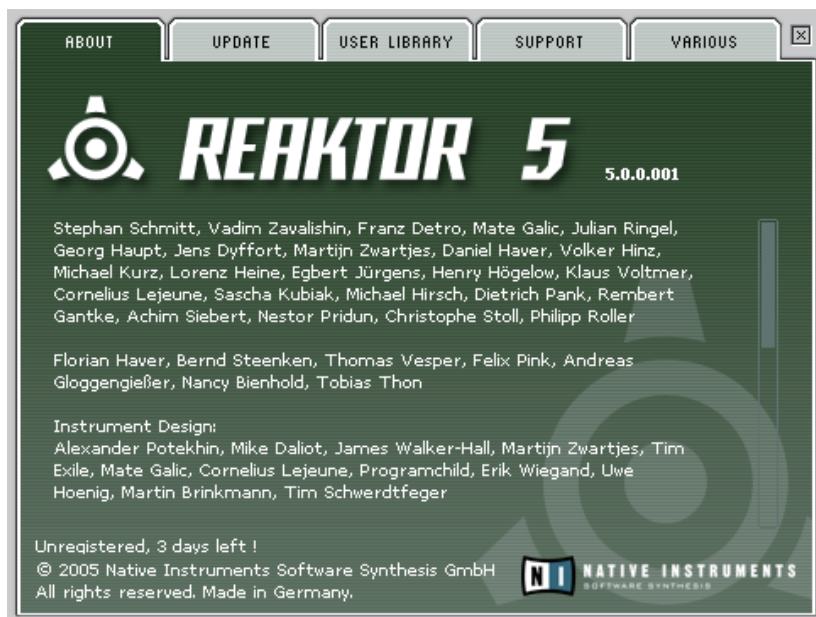
View メニューの一番下にウィンドウ名が並んでいます。これを選択すると、該当するウィンドウが最前面に出てきます。

11.6. ? メニュー

About

REAKTOR 5 に関する情報表示ウィンドウを開くコマンドです。
About タブにはソフトウェアの版番号、REAKTOR 利用許諾契約のシリアル番号が表示されます。ほかのタブには、ユーザー登録、アップデー

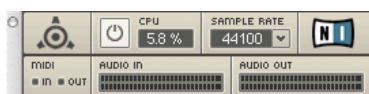
ト、NI フォーラム、技術サポート、REAKTOR User Library の各ウェブ・ページを開くためのリンクがあります。



12. REAKTOR のツールバー

REAKTOR には 3 種類のツールバーがあります。

- ・ 主ツールバー：Windows の場合 (上)、REAKTOR 全体のアプリケーション・ウィンドウの最上部にあり、REAKTOR 全体に関係する機能が集まっています。MacOS X の場合 (下) はフロート・ウィンドウとして、デスクトップ上に表示されます。



- ・ アンサンブル制御パネルのツールバー：アンサンブルに関する処理機能が集まっています。

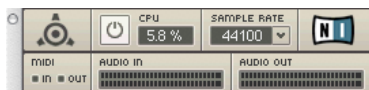


- ・ ストラクチャー・ウィンドウのツールバー：ストラクチャーの操作機能が集まっています。





注意：インストゥルメント制御パネルの最上部にあるのは「ヘッダー」であって、ツールバーとしては扱いません。

12.1. 主ツールバー





主ツールバー (上 : Windows、下 : MacOS X)


左から順に、次のようなボタン類が並んでいます。


- ・  (NI アイコン) または  (REAKTOR アイコン): REAKTOR 5 に関する情報表示ウィンドウを開きます。


- ・  **Open:** 既存のアンサンブルを開きます (Windows のみ)。


- ・  **Save:** アンサンブルを保存します (Windows のみ)。


- ・  **Undo:** 編集操作を取り消します (Windows のみ)。

- ・  **Run/Stop Audio:** オーディオ信号処理のオン / オフを切り替えます。アンサンブルの編集、音が出ている必要がなければオフにするとよいでしょう。CPU に対する負荷を軽減できます。



- ・  **CPU:** CPU に対する負荷の表示。過負荷になると「Over」という表示になります。過負荷であるとみなす限度は、環境設定の **CPU Usage** として設定できます。サウンド・カードとのやり取りや MIDI データの処理、イベント管理、画面表示、さらにはオペレーティング・システムや他のアプリケーションにも CPU が必要なので、オーディオ信号処理だけで CPU の処理時間を 100% 独占するわけにはいきません。そのため 60 ~ 80% 程度を超えると、アンサンブルを正常に処理できなくなってしまいます。どの程度までなら大丈夫か確認しておきたい場合は、負荷の状況を見ながら、声部数を徐々に増やして試してみてください。

- ・  **SAMPLE RATE:** サンプル・レートを設定します。但し、サウンド・カードが対応しているレートしか選択肢には現れません。

- ・  **AUDIO IN:** REAKTOR に入力されるオーディオ信号のレベルを表します。プレイヤーやサウンド・カードから入ってくるオーディオ信号を、**Audio In** モジュールの上 2 つのポートで測った値です。

- ・  **AUDIO OUT:** REAKTOR から出力しているオーディオ信号のレベルを表します。アンサンブルが生成するオーディオ信号を、**Audio Out** モジュールの上 2 つのポートで測った値です。

サウンド・カード、スピーカーやヘッドホン、ホスト・プログラムに対して、このレベルの信号が供給されていることになります。

- ・  **MIDI IN:** REAKTOR に対して MIDI イベントが送られてくると点灯します。なお、イベントを受け付ける MIDI 入力ポート番号は、**Audio Setup** 画面の **MIDI** ページで設定します。
- ・  **MIDI OUT:** REAKTOR から MIDI イベントを送出するときに点灯します。なお、イベントを送り出す MIDI 入力ポート番号は、**Audio Setup** 画面の **MIDI** ページで設定します。

12.2. アンサンブル制御パネルのツールバー



アンサンブル制御パネルのツールバー





- ・  **Show/Hide Panelset Bar:** パネルセット・バーの表示 / 非表示を切り替えます。
- ・  **Snapshots:** スナップショット・ウィンドウを開きます。
- ・  **Browser:** ブラウザーを開きます。
- ・  **Properties:** プロパティ・ダイアログ・ボックスを開きます。
- ・  **Ensemble Structure:** アンサンブル・ストラクチャー・ウィンドウを開きます。
- ・  **Pause/Stop Clock:** マスター・クロックを停止するボタンです。MIDI ファイルをインポートしていれば、これを押すと再生が一時停止し、もう一度押すと完全に停止して先頭に戻ります。
- ・  **Start/Restart Clock:** マスター・クロックの供給を開始するボタンです。MIDI ファイルをインポートしていれば、先頭 (一時停止中であればその位置) から再生が始まります。




-  **BPM:** マスター・クロックの速度を、1 分当たりの拍数 (BPM、Beats Per Minute) で指定します。ダブル・クリックすると数字で直接入力できるようになるほか、上下の矢印を使うと、1 BPM 単位で増減できます。
-  **MIDI Learn:** 制御パネル上のつまみ類 (ノブ、フェーダーなど) と、外部の MIDI コントローラー (モジュレーション・ホイール、ノブ、フェーダーなど) との対応を設定するためのボタンです。つまみ類をクリックして選択状態にしておき、この **MIDI Learn** ボタンを押してから、MIDI コントローラーを操作する (例えばモジュレーション・ホイールを回す) だけで対応がつきます。なお、対応を解除したい場合は、該当するつまみ類のプロパティー画面を開き、**MIDI In** プロパティーの選択を解除してください。
-  **Show/Hide Info:** これをオンにする (ボタンが点灯) と、マウス・カーソルの位置にあるオブジェクトに関する説明が現れるようになります。インストゥルメントやマクロ、モジュールに関する情報のほか、結線についても現在値が表示されます。

12.3. ストラクチャー・ウィンドウのツールバー



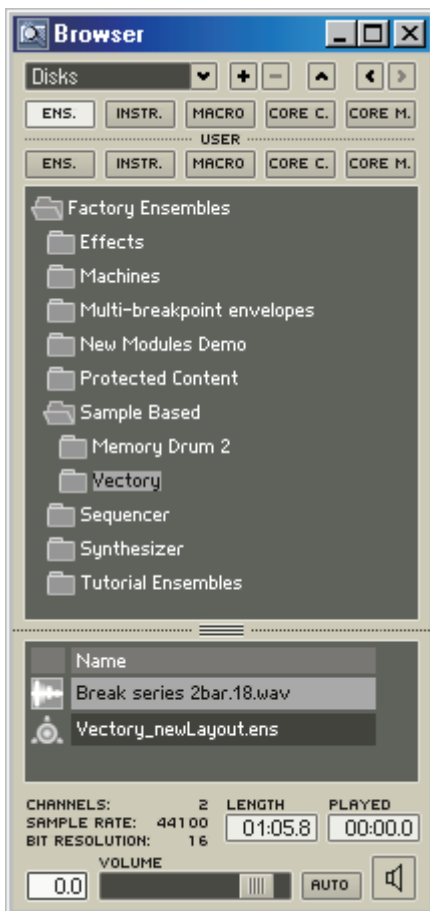
ストラクチャー・ウィンドウのツールバー

-  **Browser:** ブラウザーを開きます。
-  **Properties:** プロパティー・ダイアログ・ボックスを開きます。
-  **Ensemble Panel:** アンサンブル制御パネルを開きます。
-  **Bookmark:** ストラクチャー・ウィンドウに「しおり」をつけ、ほかのストラクチャー・ウィンドウで作業していても即座に戻れるようにします。

- ・  **Jump to Bookmark:** 左向き矢印の場合は、「しおり」のついたストラクチャー・ウィンドウに戻ります。また、「しおり」のついたウィンドウ上で作業している場合は右向き矢印になり、直前まで作業していたウィンドウに戻ることができます。
- ・  **Debug:** これはコア・セルの編集集中のみ有効です。
- ・  **Show/Hide Info:** これをオンにする（ボタンが点灯）と、マウス・カーソルの位置にあるオブジェクトに関する説明が現れるようになります。インストゥルメントやマクロ、モジュールに関する情報のほか、結線についても現在値が表示されます。

13. ブラウザー


ブラウザーを使うと、REAKTOR で使う次のようなファイルやデータを整理し、必要なものを即座に見つけることができます。



ブラウザー

- ・ オーディオ・ファイル (*.wav、*.aif、*.aiff)、アンサンブル・ファイル (*.ens)、インストゥルメント・ファイル (*.ism)、基本マクロ・ファイル (*.mdl)、コア・セル・ファイル (*.rcc)、コア・マクロ・ファイル (*.rcm)。これは次のボタンで操作します。



- ・  ボタンを押すと、よく使うフォルダーが登録され、簡単に呼び出せるようになります。
- ・ REAKTOR 付属のアンサンプル (ENS.)、インストゥルメント (INSTR.)、マクロ (MACRO)、コア・セル (CORE C.)、コア・マクロ (CORE M.)。それぞれ次のボタンで操作します。



- ・ 独自に作成したアンサンプル (ENS.)、インストゥルメント (INSTR.)、マクロ (MACRO)、コア・セル (CORE C.)、コア・マクロ (CORE M.)。それぞれ次のボタンで操作します。



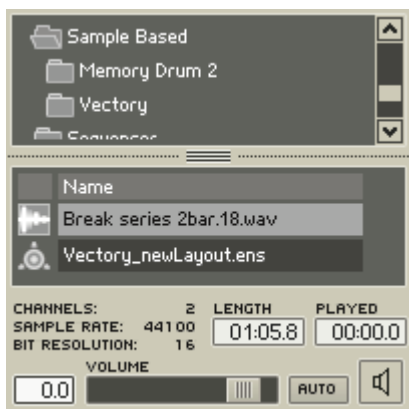
13.1. ファイルの操作

ブラウザー・ウィンドウは、主メニューの **View > Show Browser** コマンドや、ツールバーの **Show/Hide Browser** ボタンで開きます。画面は大きく 2 つの部分に分かれています。



ブラウザーの上側ペイン





上側ペインには、ディスク上のファイルを検索するボタン類のほか、**ENS.**、**INSTR.**、**MACRO**、**CORE C.**、**CORE M.** の各ボタンが 2 組あります。上段は REAKTOR に付属のシステム・ファイル用、下段は独自に作成したファイル用です。それぞれのファイルの具体的な格納場所 (パス) は、環境設定画面の **Directories** ページで設定できます。



ブラウザーの下側ペイン

一方、下側ペインには、見つけたファイル进行操作するためのつまみ類が集まっています。

ブラウザーで操作できるファイルには次のようなものがあります。

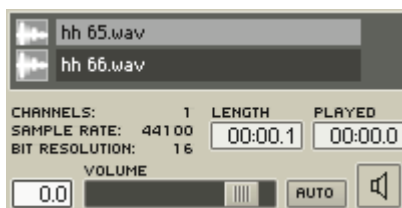
- ・  オーディオ・ファイル (*.wav、*.aif、*.aiff)。サンプル・マップ・エディター、またはインストゥルメント制御パネル (サンプラーやテープ・デッキ) 上に、ブラウザーの下側ペインからドラッグすることにより、ファイルを読み込むことができます。
- ・  アンサンプル・ファイル (*.ens)。REAKTOR のウィンドウならばどこにドラッグしても、ファイルを開くことができます。ファイルの検索には **ENS** ボタンが使えます。
- ・  インストゥルメント・ファイル (*.ism)。アンサンプル制御パネル、またはアンサンプルのストラクチャー・ウィンドウ上にドラッグすることにより、ファイルを挿入できます。ファイルの検索には **INSTR** ボタンが使えます。
- ・  基本マクロ・ファイル (*.mdl)。基本ストラクチャー・ウィンドウ上にドラッグすることにより、ファイルを読み込むことができます。ファイルの検索には **MACRO** ボタンが使えます。

- ・  コア・セル・ファイル (*.rcc)。基本ストラクチャー・ウィンドウ上にドラッグすることにより、ファイルを読み込むことができます。ファイルの検索には **CORE C.** ボタンが使えます。
- ・  コア・マクロ・ファイル (*.rcm)。コア・ストラクチャー・ウィンドウ上にドラッグすることにより、ファイルを読み込むことができます。ファイルの検索には **CORE M.** ボタンが使えます。
- ・  サンプル・マップ・ファイル (*.msp)。サンプル・マップ・エディター、またはサンプラー・モジュールの制御パネル上にドラッグすることにより、ファイルを読み込むことができます。
- ・  標準 MIDI ファイル (*.mid)。REAKTOR のウィンドウならばどこにドラッグしても、ファイルを読み込むことができます。**File > Import MIDI File** コマンドと同等の機能です。このファイルは、内蔵プレイヤーを使って再生できます。
- ・  テーブル・ファイル (*.ntf)。インストゥルメント制御パネルのテーブル表示上にドラッグすることにより、ファイルを読み込むことができます。
- ・  スナップショット・ファイル (*.ssf)。スナップショット・ウィンドウ上にドラッグすることにより、ファイルを読み込むことができます。


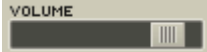

注意：ブラウザーからドラッグ操作でオブジェクト（インストゥルメント、基本マクロ、コア・セル、コア・マクロ）をストラクチャーに挿入した場合、入出力ポートに自動的に結線ができることはありません。必要に応じ、手動で結線してください。

13.2. オーディオ・ファイルの試聴

ブラウザー上でオーディオ・ファイルを試聴することができます。



ブラウザーの再生操作パネル部分

1. ブラウザー上で試聴したいオーディオ・ファイルを選択してください。
すると再生ボタンが現れ、チャンネル数やサンプル・レート、再生時間などの情報が表示されます。
2. **Play** ボタン  を押すと再生 / 停止できます。音量は **VOLUME** フェーダー  で調整してください。
3. **AUTO** ボタン  をオンにすると、オーディオ・ファイルを選択したとき、自動的に再生が始まるようになります。

14. アンサンブル

「アンサンブル」とは、REAKTOR の中で最上位階層に位置するストラクチャーのことです。制作物をアンサンブルの形でファイル (*.ens) に保存しておけば、あとでまったく同じ状態（構成するインストゥルメントやその制御パネルの設定、オーディオ入出力の接続、スナップショットなど）を再現し、作業を続けることができます。

REAKTOR の制作物は、次のような階層構造になっています。

- ・ **アンサンブル**はいくつかのインストゥルメントから成ります。
- ・ **インストゥルメント**は、他のインストゥルメントや、基本マクロ、基本モジュール、コア・セルから成ります。
- ・ **基本マクロ**は、他の基本マクロや、基本モジュール、コア・セルから成ります。
- ・ **コア・セル**はコア・マクロやコア・モジュールから成ります。
- ・ **コア・マクロ**は、他のコア・マクロや、コア・モジュールから成ります。

制御パネルも階層構造になっています。



アンサンブル制御パネル・ウィンドウ

- ・ アンサンブルには「パネル・ウィンドウ」が1つあります。



インストゥルメント制御パネル

- ・ 各インストゥルメントにはそれぞれ制御パネルがあり、アンサンブル制御パネル・ウィンドウ内に表示するか否かを個別に切り替えることができます。

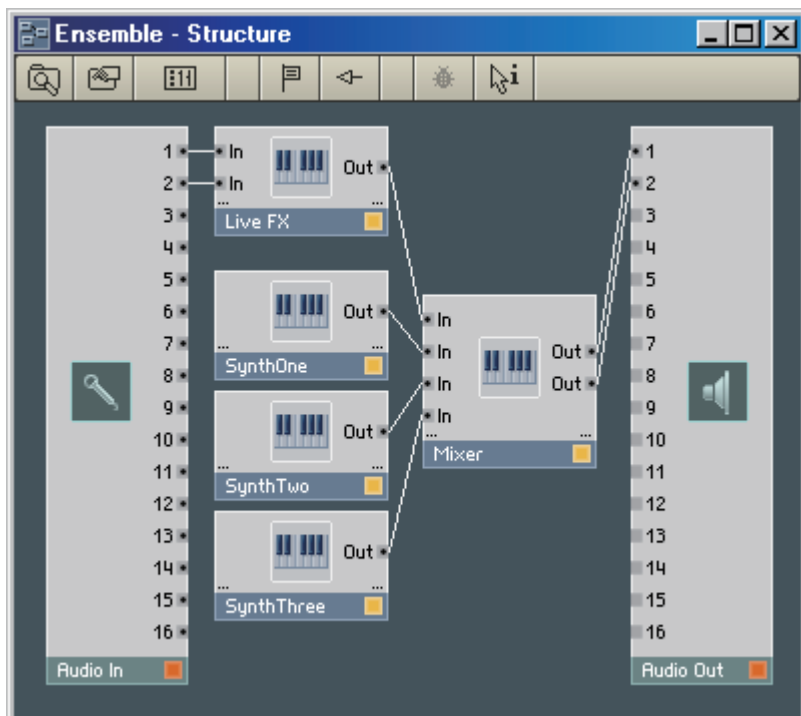


インストゥルメント制御パネル内の、マクロに対応するグループ枠

- ・ 基本マクロに対応して制御パネルのグループ枠があり、インストゥルメント制御パネル内に表示するか否かを個別に切り替えることができます。

以上は非常に簡略化した説明ですが、REAKTOR の基本的な考え方は理解できるでしょう。

14.1. アンサンブル・ストラクチャー・ウィンドウ



5つのインストゥルメントから成るアンサンブル

Ensemble Structure という名前がついたウィンドウを使うと、アンサンブル全体のストラクチャーを概観できます。アンサンブルを構成するインストゥルメントがすべてアイコン表示されているほか、サウンド・カードやプラグ・イン・ホストとのやり取りに必要な、**Audio In**、**Audio Out** モジュールがあります。上の図は、ライブ・エフェクト (**Live FX**)、3つの音源 (**SynthOne/Two/Three**)、ミキサー (**Mixer**) の、5つのインストゥルメントから成るアンサンブルです。

Audio In モジュール

Audio In モジュールは REAKTOR に対するオーディオ入力を表し、その設定は **Audio Setup** 画面の **SoundCard** ページ、**Routing** ペー

ジで行います。 **System > Audio + MIDI Settings...** コマンドで開いてください。アンサンプル・ストラクチャーには必須のモジュールで、削除はできません。

コンテキスト・メニューには次のコマンドがあります。

- ・ **Mute:** モジュールの機能を無効にします。オーディオ入力が必要ないアンサンプルについては、無効にしておくといよいでしょう。
- ・ **Properties: Audio Setup** 画面を開きます。

このモジュールには 16 のポートがあり、内蔵プレイヤー、外付けマイクなどからの信号を入力できます。実際に使えるポート数はサウンド・カードの仕様によって決まり、使えるポートは、灰色の矩形の上に黒丸がついています。一方、使えないポートには黒丸がありません。

Audio Out モジュール

Audio Out モジュールは REAKTOR からオーディオ出力を表し、その設定は **Audio Setup** 画面の **SoundCard** ページ、**Routing** ページで行います。 **System > Audio + MIDI Settings...** コマンドで開いてください。アンサンプル・ストラクチャーには必須のモジュールで、削除はできません。

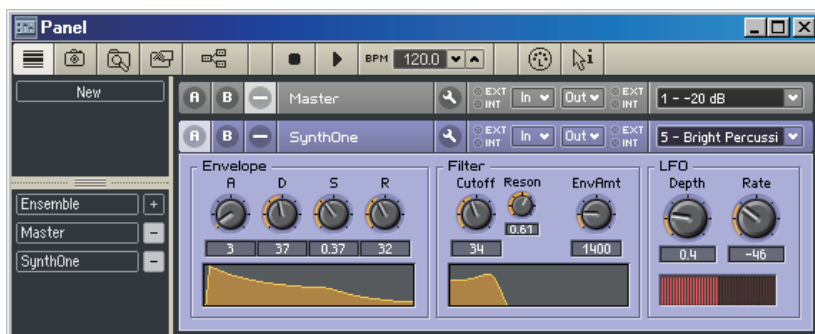
コンテキスト・メニューには次のコマンドがあります。

- ・ **Mute:** モジュールの機能を無効にします。オーディオ出力が必要ないアンサンプルについては、無効にしておくといよいでしょう。
- ・ **Properties: Audio Setup** 画面を開きます。

このモジュールには 16 のポートがあり、サウンド・カードやプラグ・イン・ホストに対して信号を供給できます。実際に使えるポート数はサウンド・カードの仕様によって決まり、使えるポートは、灰色の矩形の上に黒丸がついています。一方、使えないポートには黒丸がありません。

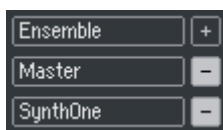
14.2. アンサンプル制御パネル・ウィンドウ


アンサンプルを構成する、各インストゥルメントの制御パネルが並んでいます。



アンサンブル制御パネル・ウィンドウ。左側はパネルセット・バー

各インストゥルメントに対応する制御パネルは、パネルセット・バーで個別に表示 / 非表示を切り替えることができます。



パネルセット・バーの表示 / 非表示は、ツールバーの **Show/Hide Panelset bar** ボタン  で切り替えます。

各インストゥルメントの制御パネルには **A**、**B2** 通りの表示様式があるほか、使わないときは最小化 (ヘッダーのみの表示) も可能です。インストゥルメント・ヘッダーにある、**A**、**B**、**-** のボタンで切り替えてください。



A、**B** の具体的な表示様式は、各つまみ類 (ノブ、フェーダーなど) ごとに、プロパティー画面の **Appearance** ページで設定します。

14.3. アンサンブルのプロパティー画面

アンサンブルのプロパティー画面は次のいずれかの手順で開くことができます。

- ・ アンサンブル制御パネル・ウィンドウの背景部分でコンテキスト・メニューを開き、**Ensemble Properties** コマンドを実行する方法。





- ・ アンサンブル・ストラクチャー・ウィンドウのツールバーにある、



Show/Hide Properties ボタンを押す方法。

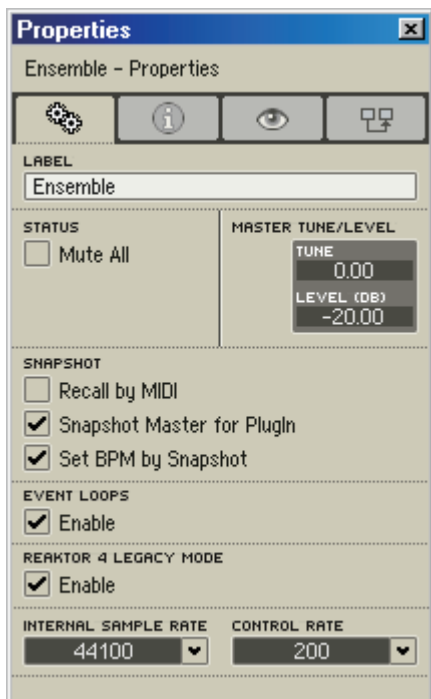
- ・ アンサンブル制御パネル・ウィンドウの背景部分をクリックし、ツールバーにある  **Show/Hide Properties** ボタンを押す方法。

- ・ アンサンブル制御パネル・ウィンドウの背景部分をクリックし、次に **View > Show Properties** コマンドを実行する方法。

プロパティ画面は **Function**、**Info**、**Appearance**、**Connection** の4つのページに分かれており、上部に並んでいる次のボタンでそれぞれ切り替えるようになっています:    .

プロパティ画面は、アンサンブル、インストゥルメント、基本マクロなどのオブジェクトを選択し直すと、それを反映して表示内容が変わります。したがって、2つのオブジェクトの設定値を見比べたい場合は、画面を開いたまま交互にオブジェクトの選択を切り替えてください。

Function ページ



アンサンブルのプロパティ画面、Function ページ

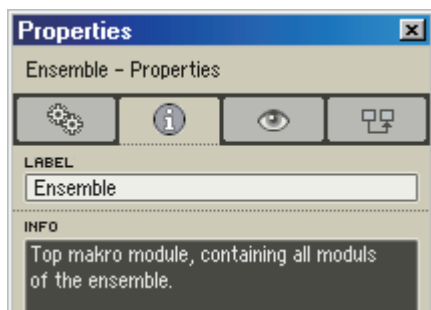
- ・ **LABEL:** 常に「Ensemble」と表示され、変更はできません。なお、インストゥルメント、基本マクロ、コア・セルなど(コア・モジュールを除く)のプロパティ画面では、この欄で名前を変更できるようになっています。
- ・ **STATUS:** **Mute All** をオンにすると、アンサンブルを構成するインストゥルメントがすべて無音になり、CPU に対する負荷も最小限になります。アンサンブル・ストラクチャー・ウィンドウを開くと、インストゥルメントには赤い「M」印、基本モジュールの入出力ポートには赤い「x」印がついています。
- ・ **MASTER TUNE/LEVEL:** **TUNE** でアンサンブル全体のピッチを調整します。半音単位(12 半音=1 オクターブ)ですが、小数も指定できるので、セント単位で微調整可能です。正值ならば高め、負値なら

ば低めになります。一方、**LEVEL (DB)** は、アンサンブル全体の音量レベル (dB 単位) を表します。音量は 6dB で倍、-6dB で半分になります。


- **SNAPSHOT: Recall by MIDI** をオンにすると、MIDI プログラム・チェンジ・メッセージ (値が 0 ~ 127) で、その値に 1 を加えた番号のスナップショットを呼び出せるようになります。例えば値が 0 ならば 1 番、1 ならば 2 番のスナップショットが呼び出されます。MIDI キーボードなどによる遠隔操作で切り替えることができるわけです。**Snapshot Master for Plugin** をオンにすると、ホスト・プログラム側からもスナップショットを呼び出せるようになります。なお、個々のインストゥルメントごとにオン / オフを切り替えることもできます。**Set BPM by Snapshot** をオンにすると、マスター・クロックの設定もスナップショットと組にして登録し、呼び出せるようになります。
- **EVENT LOOPS**: オンにすると、イベント信号のループが生じてもそれを回避しようとしなくなります。但し、処理スタックがあふれ、再生や操作が不能になる恐れがあります。オフにすると、ループが生じそうになったとき、ループの発生源を表示して続行するかどうか訊ねるようになります。REAKTOR の動作を安定させるためにも、通常はオフにしておくようお勧めします。なお、旧版で保存したファイルを読み込んだ場合は、互換性を保つためオンになっています。
- **REAKTOR 4 LEGACY MODE**: REAKTOR 5 ではイベント入力の初期設定方式が変わりましたが、これをオンにすると以前の方式で初期設定するようになります。将来の互換性を確保するためにも、特に理由がなければオフにしておくようお勧めします。
- **INTERNAL SAMPLE RATE**: REAKTOR で生成、処理するオーディオ信号のサンプル・レートを設定します。値を大きくすれば音質が上がりますが、CPU に対する負荷も高くなります。内部処理用のサンプル・レートはサブ・メニューから自由に選択できますが、選択肢はサウンド・カードやホスト・プラグ・インによって異なります。REAKTOR 側の設定が、サウンド・カードやホスト・プラグ・インの設定と一致していなければ、**Audio In**、**Audio Out** モジュールで変換されるようになります。
- **CONTROL RATE**: 1 秒間に何回の割合でイベント信号の値を更新するか、その頻度を設定します。**LFO**、**Slow Random**、**Hold**、**A to E**、**Event Smoother** など、多くの基本モジュールのイベント処理に影響します。サンプル・レートに比べると値がずっと小さいので、CPU

に対する負荷もそれほど高くありません。したがって、オーディオ信号、イベント信号のどちらを使っても結果に違いがない場合は、イベント信号で制御するようにしてください。

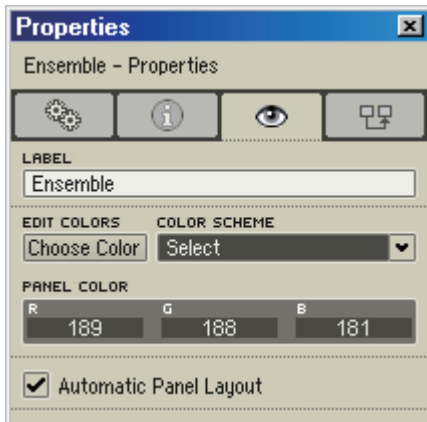
Info ページ



アンサンプルのプロパティ画面、*Info* ページ

- ・ **INFO:** このアンサンプルに関する覚え書きを残しておくことができます。アンサンプル制御パネルやストラクチャー・ウィンドウのツールバーで **Show Info**  をオンにすると、アンサンプル制御パネル・ヘッダー上にマウス・カーソルを持っていったとき、この覚え書きが表示されるようになります。

Appearance ページ



アンサンプルのプロパティー画面、*Appearance* ページ

- **EDIT COLORS:** **Choose Color** ボタンを押すとカラー・パレット画面が現れ、アンサンプル制御パネル・ヘッダーの色を設定できます。本来はパネルの色を指定する機能なのですが、アンサンプルの場合、パネル・ヘッダーだけで本体は存在しません。
- **COLOR SCHEME:** **Save as Custom** を選択すると、現在の色設定を、独自の色遣いとして登録できます。他のアンサンプルと色遣いを揃えたい場合、あらかじめ登録しておいた色遣いを、**Set to Custom** で呼び出してください。なお、登録しておける色遣いはひと通りに限ります。**Set to Default** を選択すると、REAKTOR 標準の色遣い (灰色のパネル、橙色のインジケーター) になります。
- **PANEL COLOR:** パネル・ヘッダーの色は、上述のカラー・パレット画面を使うほか、3 原色 (赤、緑、青) の混合割合でも指定できます。3 原色とも 0 ならば黒、256 ならば白になります。
- **Automatic Panel Layout:** 初期状態ではオンになっており、アンサンプル制御パネル・ウィンドウ内の各パネルが自動的に整列表示されます。オフにすると、どの位置にでも自由に動かせるようになります。

Connection ページ



アンサンブルのプロパティ画面、**Connection** ページ

- ・ **MIDI IN DEVICE**: MIDI 入力機器が複数ある場合に、どこからの MIDI メッセージを待ち受けるかを設定します。**Audio Setup** 画面の **MIDI** ページで有効とした MIDI 機器の中から、アンサンブルごとを選択できるようになっています。どの MIDI 機器からのメッセージでも受信できるよう、通常は **All** にしておいても構いません。何か問題があって、特定の MIDI 機器からのメッセージを無視したいような場合に、適宜設定を変えてください。
- ・ **CHANNEL**: MIDI 入力を待ち受けるチャンネル番号を指定します。これ以外のチャンネルの MIDI メッセージは無視されます。
- ・ **MORPH**: スナップショットのモーフィング機能のオン / オフを切り替えます。

- ・ **CONTROLLER**: モーフィングに使うコントローラー番号を指定します。
- ・ **OSC SOURCE**: OSC データをどこから受け取るか指定します。
- ・ **OSC TARGET**: OSC データの送信先を指定します。
- ・ **OSC CONNECTIONS**: 有効な OSC 接続が列挙されています。

External Sync: Sync Clock、1/96 Clock モジュールで、REAKTOR 内蔵クロックを使うか、外部クロック源を MIDI 経由で受け取って使うか、を切り替えます。また、**Start/Stop** モジュールを、外部からの MIDI-Start/Stop メッセージで制御するかどうか、切り替える働きもあります。オンにすると、主ツールバーの **BPM** 欄の値は無視し、外部クロックに同期して動作するようになります。

注意: **Settings** メニューでも切り替え可能です。

15. インストゥルメント

REAKTOR の「インストゥルメント」は独自の内部ストラクチャーや制御パネルを持ち、MIDI 処理やスナップショット管理の機能もあります。アンサンプル・ストラクチャー・ウィンドウ上で見ると、青い帯上に名前が表示され、キーボードをかたどったアイコンがついています。



インストゥルメント・オブジェクト

「インストゥルメント」は、他のインストゥルメントや、基本マクロ、基本モジュール、コア・セルから成ります。上の図では「Instrument」という名前が表示されていますが、変えることもできます。

15.1. インストゥルメントをアンサンプルに追加する手順

REAKTOR に付属のインストゥルメント、または独自に作成したインストゥルメントを、次の手順でアンサンプルに追加できます。



- ・ ストラクチャー・ウィンドウの背景部分でコンテキスト・メニューを開き、該当するインストゥルメントを選択、追加する方法。一般にはアンサンプルのストラクチャーに対して追加しますが、インストゥルメントのストラクチャーに対して別のインストゥルメントを追加することも可能です。
- ・ 同様にコンテキスト・メニューを開き、**Insert Instrument** コマンドを実行する方法。
- ・ ブラウザーの **INSTR.** ボタン (上はシステム付属、下は独自作成) を押して該当するインストゥルメントを検索し、ストラクチャー上の追加したい位置までドラッグする方法。
- ・ ブラウザーのファイル検索機能 (最上部) を使ってインストゥルメントを検索し、同様にドラッグする方法。

システム・ライブラリーには、音声合成やエフェクトなど、よく使うインストゥルメントが揃っています。独自のインストゥルメントを作成しようという場合は、「_New」で始まる名前の、空のインストゥルメントをまず読み込んでください。

インストゥルメントを読み込むという処理では、ファイルの中身を複製しています。したがって、あとでファイルを変更しても、読み込み済みのインストゥルメントは影響を受けません。同様に、REAKTOR 上でインストゥルメントを編集しても、ファイルの中身は元のままです。ファイル自身を変更したい場合は、アンサンブル内に読み込んだ (複製した) インストゥルメントを編集した後、**Save Instrument As...** コマンドで上書き保存してください。

注意: インストゥルメントのストラクチャーに対して、別のインストゥルメントを追加することも可能です。

15.2. ポート

インストゥルメントではポートの配置が可変で、入出力端子系のモジュール (  など) を追加することにより、種類や数を自由に決めることができます。

ポートをを介してインストゥルメント外とやり取りする信号は、単声のものに限ります。したがって多声のインストゥルメントの場合、**Audio Voice Combiner** モジュールを出力ポートの前段に置いて変換する必要があります。

15.3. コンテキスト・メニュー

アンサンブル・ストラクチャー・ウィンドウ内で、インストゥルメント上でコンテキスト・メニューを開くと、次のようなコマンドが並んでいます。

- ・ **Mute:** インストゥルメントを無音にします。
- ・ **Solo:** このインストゥルメントの出力を、**Audio Out** モジュールに直接接続します。信号経路上、このインストゥルメントの上流に位置するインストゥルメントはすべて有効になりますが、それ以外は無音 (ミュート) 状態になります。例えば、Synth インストゥルメントの出力信号を、コーラス・エフェクト、コンプレッサーを経由して **Audio Out** モジュールにつないでいるとしましょう。ここでコーラス・エフェクトを **Solo** モードにすると、ここから直接 **Audio Out** モジュールに信号が伝わり、コンプレッサーの効果がかかっていない信号が最終的に出力されることになります。

- ・ **Cut:** インストゥルメントをクリップボードに複製し、ストラクチャー上から削除します。これを **Paste** コマンドで別の位置に貼り付けることができます。
- ・ **Copy:** 同様にクリップボードに複製しますが、ストラクチャー上からは削除しません。
- ・ **Duplicate:** インストゥルメントを複製し、同じストラクチャー上に貼り付けます。**Copy**、**Paste** の 2 つのコマンドを実行したのと同様です。
- ・ **Delete:** インストゥルメントをストラクチャー上から削除します。
- ・ **Save Instrument As...:** インストゥルメントに名前をつけてファイル (*.ism) に保存します。通常は、独自に作成したファイルを置く、専用のフォルダーに保存することになるでしょう。
- ・ **Structure:** このインストゥルメントのストラクチャー表示に切り替えます。なお、インストゥルメントのアイコン上をダブル・クリックしても同じです。
- ・ **Structure Window:** 別のウィンドウを開いて、このインストゥルメントのストラクチャーを表示します。したがって複数のウィンドウが開いていることになります。なお、**Alt** キーを押しながらダブル・クリックしても同じ機能になります。
- ・ **Properties:** インストゥルメントのプロパティー画面を開きます。詳しくは「インストゥルメントのプロパティー画面」の節を参照してください。









15.4. インストゥルメント・ヘッダー

インストゥルメント・ヘッダーには次のようなつまみ類が並んでいます。



インストゥルメント・ヘッダー

- ・ **A B** **A、B:** 制御パネルの表示様式 (**A**)、(**B**) を切り替えるボタンです。それぞれの具体的な表示様式は、つまみ類 (ノブ、フェーダーなど) ごとに、**A**、**B**、**AB**、**Visible** の各プロパティーであらかじめ設定しておきます。





-  **Minimize:** 制御パネルを最小化表示に切り替えます。もう一度押すと元に戻ります。
-  : プロパティー画面の **LABEL** 欄で指定した名前です。
-  **Lock/Unlock Panel:** 制御パネルのロック / ロック解除を切り替えます。ロックすると各つまみ類の配置が変更されなくなります。通常使用時はロックしておいてください。ロックを解除すると、つまみ類の配置を自由に調整できるようになりますが、値の調節はできません。制御パネルの作成中は解除して作業することになります。
-  **EXT/INT、In/Out:** MIDI 信号の送受信状況を表示するランプです。内蔵 / 外付け MIDI 機器との間で信号の入力 / 出力があると点灯します。なお、表示の対象となる MIDI ポートは、**Audio Setup** 画面の **MIDI** ページで設定します。
-  **In、Out:** インストゥルメントの入出力接続を設定します。
-  **Snapshot:** あらかじめ登録してあったスナップショットを呼び出します。
-  **VOICES:** 同時発音声部数を表します。なお、プロパティー画面の **Function** ページでも設定できます。
-  **UNISON:** 斉奏 (ユニゾン) で鳴らせる声部の最大数を表します。パラメーターをわずかに変えて同時に鳴らすことにより、単に 1 つの声部を再生した場合とは異なる、厚みのある響きを実現できます。なお、パラメーターを変化させる幅は **UNISON SPRD**、声部の最小数は **MIN UNISON V** として、プロパティー画面の **Function** ページで設定します。

注意: **UNISON** の値は、スナップショットと組にして保存できます。逆に言うと、スナップショットごとに設定を変えて管理できるようになります。これに対して **VOICES** は、どのスナップショットに対しても同じ値が適用されます。

15.5. インストゥルメントのプロパティー画面

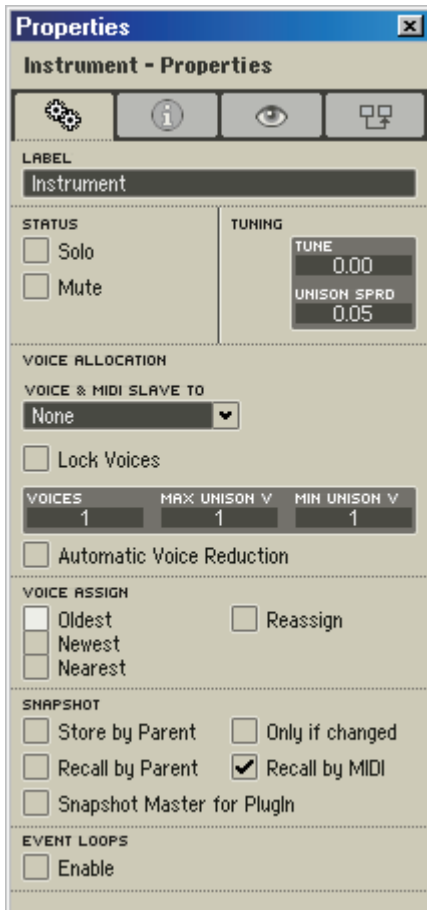
インストゥルメントのプロパティー画面は次のいずれかの手順で開くことができます。

- ・ 青い帯上に表示された名前の部分、あるいはパネル・ヘッダーに表示された名前の部分をダブル・クリックする方法。
- ・ インストゥルメント・アイコン上、あるいはパネル・ヘッダーに表示された名前の部分でコンテキスト・メニューを開き、**Properties** コマンドを実行する方法。
- ・ アンサンプル制御パネル・ウィンドウやストラクチャー・ウィンドウでインストゥルメントを選択し、**View > Show Properties** コマンドを実行する方法。

プロパティー画面は **Function**、**Info**、**Appearance**、**Connection** の4つのページに分かれており、上部に並んでいる次のボタンでそれぞれ切り替えるようになっています：   。

プロパティー画面は、アンサンプル、インストゥルメント、基本マクロなどのオブジェクトを選択し直すと、それを反映して表示内容が変わります。したがって、2つのオブジェクトの設定値を見比べたい場合は、画面を開いたまま交互にオブジェクトの選択を切り替えてください。

Function ページ



インストゥルメントのプロパティー画面、Function ページ

- ・ **LABEL:** インストゥルメント名を表し、ここで変更も可能です。インストゥルメント制御パネル・ヘッダーにはこの名前が表示されます。
- ・ **STATUS:**
 - ・ **Solo:** このインストゥルメントの出力を、**Audio Out** モジュールに直接接続します。信号経路上、このインストゥルメントの上流に位置するインストゥルメントはすべて有効になりますが、それ

以外は無音 (ミュート) 状態になります。コンテキスト・メニューの **Solo** も参照してください。

- ・ **Mute:** 信号経路上、このインストゥルメントの上流に位置するインストゥルメントを、すべて無音にします。ストラクチャー・ウィンドウを開くと、状態 LED には赤い「M」印、入出力ポートには赤い「x」印がついています。
- ・ **TUNING:**
 - ・ **TUNE:** アンサンブル全体のピッチを基準として、インストゥルメントのピッチを調整します。半音単位 (12 半音 = 1 オクターブ) ですが、小数も指定できるので、セント単位で微調整可能です。正值ならば高め、負値ならば低めになります。インストゥルメントごとにわずかにずらして音に厚みを加える場合、0.05 (すなわち 5 セント) 程度が適当でしょう。
 - ・ **UNISON SPRD:** 斉奏 (ユニゾン) で鳴らす各声部のピッチをどの程度ずらすか、を表します (もちろん 2 声部以上を割り当てておく必要があります)。TUNE と同様に半音単位で表します。通常は 0.05 (すなわち 5 セント) 程度が適当でしょう。
- ・ **VOICE ALLOCATION:** 声部をどのように割り当てるか、インストゥルメントごとに設定できます。
 - ・ **VOICE & MIDI SLAVE TO:** 声部の割り当てや MIDI 入力の設定を、同じアンサンブルに属する別のインストゥルメントから制御したい場合に、その制御元インストゥルメントを指定します。
 - ・ **Lock Voices:** オンにすると、**VOICES**、**MAX UNISON V**、**MIN UNISON V** の設定を変更できなくなります。変更の必要があるときだけオフにしてください。
 - ・ **VOICES:** 同時発音声部数を表します。このインストゥルメントに属する、多声モジュールすべてに適用され、個々のモジュールの **Mono** プロパティーは変更できません。
 - ・ **MAX UNISON V:** 斉奏 (ユニゾン) で鳴らせる声部の最大数を表します。なお、声部の最小数は次の **MIN UNISON V**、各声部のピッチをずらす幅は **UNISON SPRD** で指定します。
 - ・ **MIN UNISON V:** 斉奏 (ユニゾン) で鳴らせる声部の最小数を表します。

VOICES、**MAX UNISON V**、**MIN UNISON V** はそれぞれ独立に設定できます。**VOICES** は、このインストゥルメントで同時に発音できる声部数を表します。斉奏しない場合は、**MAX UNISON V**、**MIN**

UNISON Vとも1と設定してください。一方、1つの音符をいくつかの声部で同時に鳴らす場合は、その声部数の最大 / 最小を指定します。

例えば、**VOICES**、**MAX UNISON V**、**MIN UNISON V**をそれぞれ24/1/1と設定すれば、同時に24の音符を鳴らして和音を作ることができますが、斉奏はしないことになります。一方、24/3/3と設定すれば、8つまでの音符を、それぞれ3つの声部で鳴らすことができます。24/3/2ならば、8つの音符をそれぞれ3つの声部で鳴らせますが、音符数が12に増えても、2声部ずつを割り当てることができます。

このように**MAX UNISON V**と**MIN UNISON V**が等しくない場合、音符の数に合わせて、何声部ずつ割り当ててを自動的に切り替えるようになります。24/3/2の設定の場合、音符が8個以下ならば3声部、8個を超えれば2声部というように調整されるわけです。

- ・ **Automatic Voice Reduction**: オンにすると、CPU に対する負荷が所定の比率 (環境設定の **CPU Usage**) を超えたとき、自動的に同時発音声部数を減らすようになります。処理能力に応じて声部数を調整できることになります。
- ・ **VOICE ASSIGN**: 割り当てられた声部数では足りなくなったとき、新しい音を鳴らす代わりにどの声部の音を捨てるか、判断する基準を表します。次の3つから選択してください。
 - ・ **Oldest**: 最も古くから鳴っていた声部の音を捨てます。これが最も普通の設定でしょう。
 - ・ **Newest**: 最も新しく鳴り始めた声部の音を捨てます。ドローンなどの持続音の上にメロディーが演奏される場合に適用すると、持続音が停まってしまうことはありません。
 - ・ **Nearest**: 最も高さの近い音を捨てます。多声発音の機能を使ってポルタメントを実現している場合に有効です。
- ・ **Reassign**: 同じ音を繰り返して演奏したとき、同じ声部を再利用するか、別の声部に割り当て直すか、を設定します。限られた声部数を有効に使いたい場合はオンにしておくといよいでしょう。ピアノで同じキーを連打するとき、2本の指を交互に使う演奏法がありますが、これと同じ考え方です。

次の **SNAPSHOT** については、第20章「スナップショット」も参照してください。

- ・ **Recall by Parent** をオン、**Store by Parent** をオフにすると、上位階層オブジェクト (アンサンブルまたは別のインストゥルメント) のスナップショットを呼び出すことにより、このインストゥルメン

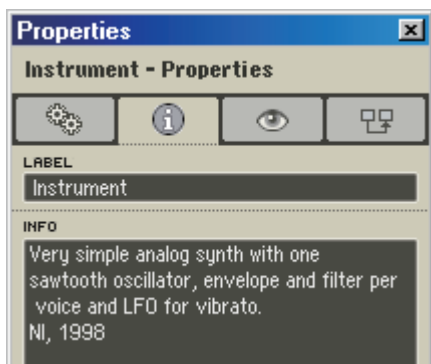
トのスナップショットも自動的に呼び出されるようになります。例えば、あるインストゥルメントのスナップショット「iSnap1」を保存し、次にこの上位に当たるアンサンブルのスナップショット「eSnap1」を保存したとします。すると、「eSnap1」を呼び出すだけで、自動的に「iSnap1」も呼び出されます。

- **Recall by Parent、Store by Parent** とともにオンにすると、上位階層オブジェクトのスナップショットを保存し、または呼び出すことにより、このインストゥルメントのスナップショットも自動的に保存され、または呼び出されるようになります。例えば、あるインストゥルメントのスナップショットとして保存する内容を作成し、次にこの上位に当たるアンサンブルのスナップショット「Snap1」を保存したとします。すると、「Snap1」を呼び出すだけで、インストゥルメントについても自動的にスナップショットが呼び出されます。
- **Recall by Parent** をオフ、**Store by Parent** をオンにすると、上位階層オブジェクトのスナップショットを保存することにより、このインストゥルメントのスナップショットも自動的に保存されるようになります。例えば、あるインストゥルメントのスナップショットとして保存する内容を作成し、次にこの上位に当たるアンサンブルのスナップショット「Snap1」を保存したとします。するとインストゥルメントについても同じ「Snap1」という生でスナップショットが保存されます。但し、アンサンブルの側でスナップショットを呼び出しても、インストゥルメントについて自動的にスナップショットが呼び出されるわけではありません。
- **Only if changed** と **Store by Parent** をオンにすると、上位階層オブジェクトのスナップショットを保存するとき、このインストゥルメントについては、前回と内容が変わっている場合に限り自動的に同じ名前で保存されるようになります。スナップショット履歴に並ぶ項目数が無駄に多くなりません。
- **Recall by MIDI** をオンにすると、MIDI プログラム・チェンジ・メッセージ (値が 0 ~ 127) で、その値に 1 を加えた番号のスナップショットを呼び出せるようになります。例えば値が 0 ならば 1 番、1 ならば 2 番のスナップショットが呼び出されます。MIDI キーボードなどによる遠隔操作で切り替えることができるわけです。
- **Snapshot Master for Plugin** をオンにすると、ホスト・プログラム側からもスナップショットを呼び出せるようになります。但しこのホスト・プログラムは 1 つに限ります。なお、アンサンブルに属する


インストゥルメントについて、一括してこの設定をすることも可能です。

- ・ **EVENT LOOPS:** オンにすると、イベント信号のループが生じてもそれを回避しようとしなくなります。但し、処理スタックがあふれ、再生や操作が不能になる恐れがあります。オフにすると、ループが生じそうになったとき、ループの発生源を表示して続行するかどうか訊ねるようになります。REAKTOR の動作を安定させるためにも、通常はオフにしておくようお勧めします。

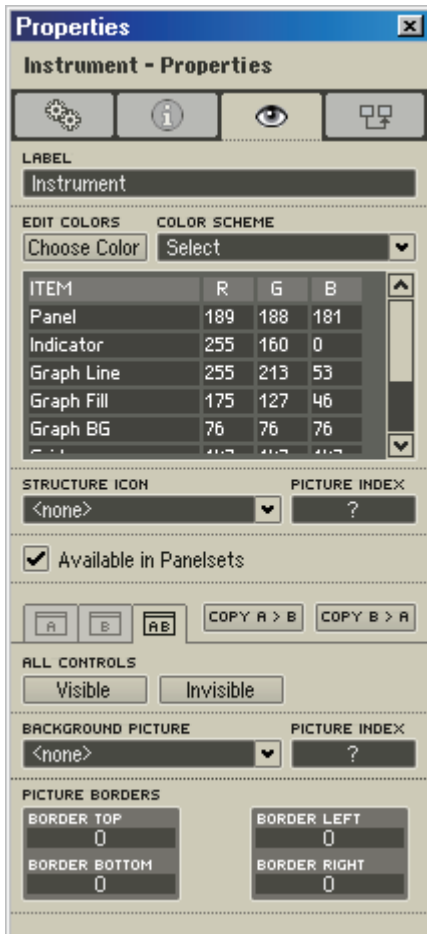
Info ページ



インストゥルメントのプロパティー画面、Info ページ

- ・ **INFO:** このインストゥルメントに関する覚え書きを残しておくことができます。アンサンブル制御パネルやストラクチャー・ウィンドウのツールバーで **Show Info**  をオンにすると、インストゥルメント制御パネル・ヘッダー上にマウス・カーソルを持っていったとき、この覚え書きが表示されるようになります。

Appearance ページ



インストールメントのプロパティ画面、*Appearance* ページ

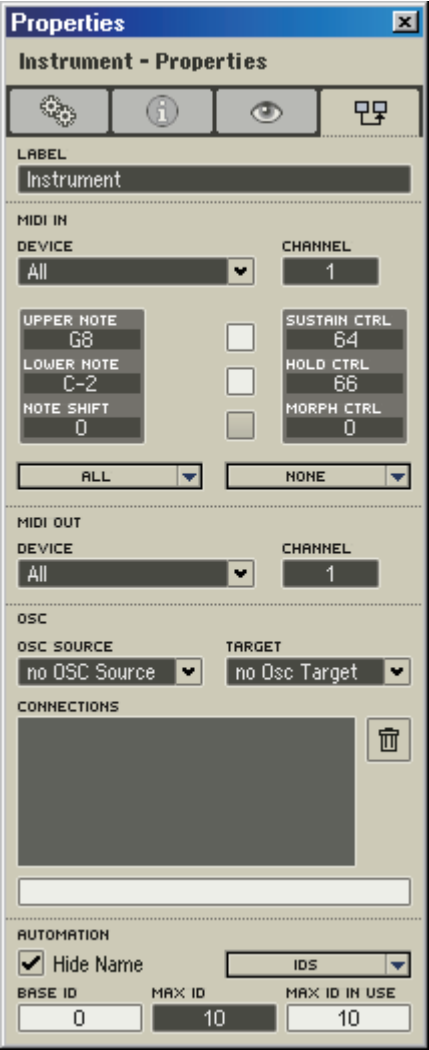
- ・ **EDIT COLORS: Choose Color** ボタンを押すとカラー・パレット画面が現れ、この下の欄で選択状態になっている項目の色を設定できます。
- ・ **COLOR SCHEME: Save as Custom** を選択すると、現在の色設定を、独自の色遣いとして登録できます。他のアンサンブルと色

遣いを揃えたい場合、あらかじめ登録しておいた色遣いを、**Set to Custom** で呼び出してください。なお、登録しておける色遣いはひと通りに限ります。**Set to Default** を選択すると、REAKTOR 標準の色遣い (灰色のパネル、橙色のインジケーター) になります。

- ・ **ITEM:** インストゥルメント制御パネル上で、色を変更できる項目が並んでいます。上述のカラー・パレット画面を使うほか、3 原色 (赤、緑、青) の混合割合でも指定できます。3 原色とも 0 ならば黒、256 ならば白になります。具体的には次のような項目があります。
 - ・ **Panel:** 制御パネルの背景色。但し別に **BACKGROUND PICTURE** として背景画像が指定されている場合を除きます。
 - ・ **Indicator:** つまみ類 (ノブ、フェーダー、ボタンなど) のインジケーター部分の色。
 - ・ **Graph Line:** **Audio Table**、**Event Table** モジュールの曲線、**XY** モジュールのカーソル、フィルター特性曲線やエンベロープ曲線の色。
 - ・ **Graph Fill:** **Audio Table**、**Event Table** モジュールの曲線内、**XY** モジュールのオブジェクト、フィルター特性曲線内やエンベロープ曲線内を埋める色。
 - ・ **Graph BG:** **Audio Table**、**Event Table**、**XY** モジュール、フィルター特性曲線やエンベロープ曲線の背景部分の色。
 - ・ **Grid:** **Audio Table**、**Event Table** モジュールの格子の色。
 - ・ **2D Table Min:** **Audio Table**、**Event Table** モジュールを 2 次元モードで使う場合の、最小値の表示色。
 - ・ **2D Table Max:** 同じく、最大値の表示色。
 - ・ **2D Table Default:** 同じく、デフォルト値の表示色。
- ・ **STRUCTURE ICON:** ストラクチャー・ウィンドウに表示する際のアイコン画像。特に指定しなければ、キーボードをかたどったアイコンになります。
- ・ **PICTURE INDEX:** **STRUCTURE ICON** として、複数の画像を縦または横に並べたビットマップを指定した場合、何番目の画像を適用するか、その番号を指定します。なお、画像の個数は **NUM ANIMATIONS** プロパティとして設定します。
- ・ **Available in Panelsets:** オンにすると、パネルセット・バーの下段にこのインストゥルメント名が表示されるようになります。アンサンブル制御パネル・ウィンドウに表示するか否か、ここで切り替えることができます。

- ・ **A、B、AB:** 制御パネルは **A、B** の 2 つがあって、随時切り替えながら使えるようになっていますが、このインストールメントをどちらの制御パネルに表示するか、ここで指定します。以下に説明する **ALL CONTROLS、PICTURE BORDERS** の設定は、ここで指定した制御パネルについてのみ適用されます。
 - ・ **COPY A > B、COPY B > A:** 一方の制御パネルの中身を、そのままもう一方にコピーするためのボタンです。
 - ・ **ALL CONTROLS: Visible/Invisible:** 制御パネル上のつまみ類を一括して、表示 / 非表示を切り替えることができます。
 - ・ **BACKGROUND PICTURE:** 制御パネルの背景に表示する画像を指定します。**A** と **B** で別の画像を設定できます。
 - ・ **PICTURE INDEX:** 背景画像として、複数の画像を縦または横に並べたビットマップを指定した場合、何番目の画像を適用するか、その番号を指定します。なお、画像の個数は **NUM ANIMATIONS** プロパティーとして設定します。
- PICTURE BORDERS:** 制御パネルの上下左右に確保する余白を、ピクセル単位で指定します。**A** と **B** でそれぞれ異なる値を指定できます。なお、格子を表示する関係上、4 の倍数で指定してください。

Connection ページ



Properties

Instrument - Properties

LABEL

Instrument

MIDI IN

DEVICE: All CHANNEL: 1

UPPER NOTE: G8 LOWER NOTE: C-2 NOTE SHIFT: 0

SUSTAIN CTRL: 64 HOLD CTRL: 66 MORPH CTRL: 0

ALL NONE

MIDI OUT

DEVICE: All CHANNEL: 1

OSC

OSC SOURCE: no OSC Source TARGET: no Osc Target

CONNECTIONS

AUTOMATION

☒ Hide Name IDS

BASE ID: 0 MAX ID: 10 MAX ID IN USE: 10

インストールメントのプロパティ画面、**Connection** ページ

- ・ MIDI IN:

- ・ **DEVICE:** MIDI 入力機器が複数ある場合に、どこからの MIDI メッセージを待ち受けるかを設定します。**Audio Setup** 画面の **MIDI** ページで有効とした MIDI 機器の中から、アンサンブルごとに選択できるようになっています。どの MIDI 機器からのメッセージでも受信できるよう、通常は **All** にしておいても構いません。何か問題があって、特定の MIDI 機器からのメッセージを無視したいような場合に、適宜設定を変えてください。
- ・ **CHANNEL:** MIDI 入力を待ち受けるチャンネル番号を指定します。これ以外のチャンネルの MIDI メッセージは無視されます。
- ・ **UPPER NOTE、LOWER NOTE:** このインストゥルメントで認識できる音域を表します。この音域外のノート番号が与えられても無視します。この機能を利用して、複数のインストゥルメントを音域に応じて自動的に切り替えながら演奏することも可能です。
- ・ **NOTE SHIFT:** 移調して演奏する場合に、その移調幅を半音単位で指定します。例えば「-12」とすれば、1 オクターブ低い音が出ます。
- ・ **SUSTAIN CTRL:** サスティン・ペダルとして使う MIDI コントローラー番号を指定します。標準値は 64 です。MIDI 機器の製造元によっては、「ホールド」、「ダンパー・ペダル」などという名前になっています。サスティン・ペダルを踏んでいる間は、キーを放してもその音が鳴り続けます。この欄の左側にあるチェック・ボックスをオンにすると機能が有効になります。
- ・ **HOLD CTRL:** ホールド・ペダルとして使う MIDI コントローラー番号を指定します。標準値は 66 です。MIDI 機器の製造元によっては、「ソステヌート」などという名前になっていることもあります。ホールド・ペダルを踏んでいる間に叩いたキーの音は、キーを放しても鳴り続けます。この欄の左側にあるチェック・ボックスをオンにすると機能が有効になります。
- ・ **MORPH CTRL:** スナップショットのモーフィング処理に使う MIDI コントローラー番号を指定します。この欄の左側にあるチェック・ボックスをオンにすると機能が有効になります。
- ・ **ALL:** インストゥルメントに関係するすべてのつまみ類について、このメニューから選択した設定を適用します。
- ・ **NONE:** すべてのつまみ類について、このメニューから選択した設定の適用を除外します。

個々の選択肢の内容については、第 18 章「制御パネルの編集」を参照してください。

- **MIDI OUT:**

- **DEVICE:** MIDI 出力機器が複数ある場合に、MIDI メッセージをどこに送るかを設定します。**Audio Setup** 画面の **MIDI** ページで有効とした MIDI 機器の中から選択できるようになっています。

- **CHANNEL:** MIDI メッセージの出力先チャンネル番号を指定します。

- **OSC:**

- **OSC SOURCE:** OSC データをどこから受け取るか指定します。**OSC Setup** ウィンドウに登録されているコンピューターが、選択肢として列挙されます。
- **TARGET:** OSC データの送信先を指定します。
- **CONNECTIONS:** インストゥルメントがストラクチャー階層上のどの位置にあるか、を表示します。例えば「Ensemble」という名前のアンサンブルに属する「Synth」というインストゥルメントの場合、「Ensemble/Synth」となります。

16. 基本マクロ

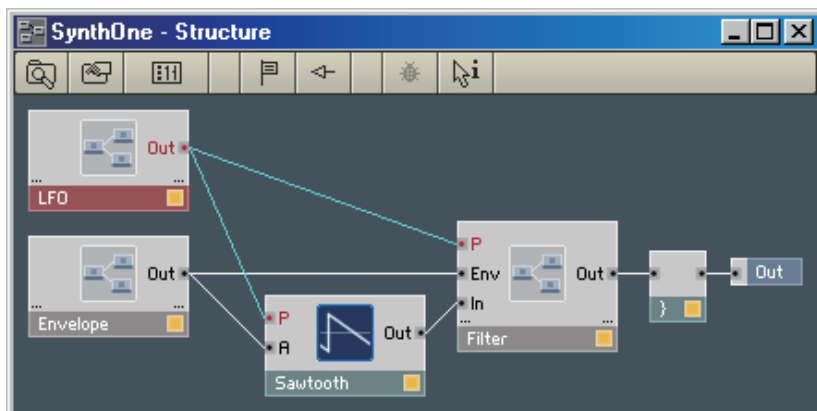
16.1. 「基本マクロ」の目的

「基本マクロ」はインストゥルメントと同じく、モジュールその他を組み合わせるとまとまりの機能を実現するストラクチャーです。但しMIDI 管理、制御パネル、スナップショットの機能はありません。ストラクチャー・ウィンドウ上で見ると、灰色の帯上に名前が表示され、3つのモジュールを結線した様子をかたどったアイコンがついています。





「マクロ」オブジェクト

あるひとまとまりの機能を単位としてまとめることにより、複雑なストラクチャーを階層化して管理しやすくするのが目的です。ある程度以上の規模のストラクチャーは、基本マクロを組み合わせる形で作成するようにしてください。機能単位が明確であれば、他の用途に再利用するのも容易です。



チャー内に読み込んだ (複製した) マクロを編集した後、**Save Macro As...** コマンドで上書き保存してください。

16.3. ポート

マクロではポートの配置が可変で、入出力端子系のモジュール (、  など) を追加することにより、種類や数を自由に決めることができます。

入出力端子系のモジュールは、上位階層のストラクチャー画面で見ると、ポートの形で現れます。あるマクロ・ストラクチャー上に入力および出力のモジュールを置き、背景部分をダブル・クリックして上位階層表示に切り替えると、マクロの左側に入力ポート、右側に出力ポートが表示されます。信号はこの入力ポートを経由してマクロ内部に入り、処理結果が出力ポートを経由して外に出てくることになります。

上位階層のストラクチャー画面でポートを追加することも可能です。Windows ならば **Ctrl** キー、MacOS X ならば **command** キーを押しながら、接続元 / 先モジュールのポート上をクリックしてそのままドラッグし、入力ポートを追加するのであればマクロ・アイコンの左側、出力ポートであれば右側の境界線近くまで持って行ってください。するとポートが現れるので、マウス・ボタンを放します。ポート名は、マクロ名または接続元 / 先モジュール名と同じになります。

16.4. コンテキスト・メニュー

マクロのコンテキスト・メニューには、次のようなコマンドが並んでいます。





- ・ **Mono**: オンにすると単声信号を処理するモードになります。詳しくは **STATUS** プロパティの項を参照してください。
- ・ **Mute**: マクロの機能を無効にします。詳しくは **STATUS** プロパティの項を参照してください。
- ・ **Cut**: マクロをクリップボードに複製し、ストラクチャー上から削除します。これを **Paste** コマンドで別の位置に貼り付けることができます。
- ・ **Copy**: 同様にクリップボードに複製しますが、ストラクチャー上からは削除しません。

- ・ **Duplicate**: マクロを複製し、同じストラクチャー上に貼り付けます。**Copy**、**Paste** の2つのコマンドを実行したのと同様です。
- ・ **Delete**: マクロをストラクチャー上から削除します。
- ・ **Save Macro As...**: マクロに名前をつけてファイル (*.mdl) に保存します。通常は、独自に作成したファイルを置く、専用のフォルダーに保存することになるでしょう。
- ・ **Structure**: このマクロのストラクチャー表示に切り替えます。なお、マクロのアイコン上をダブル・クリックしても同じです。
- ・ **Structure Window**: 別のウィンドウを開いて、このマクロのストラクチャーを表示します。したがって複数のウィンドウが開いていることになります。なお、**Alt** キーを押しながらダブル・クリックしても同じ機能になります。
- ・ **Properties**: マクロのプロパティ画面を開きます。詳しくは「マクロのプロパティ画面」の節を参照してください。

16.5. マクロのプロパティ画面

マクロのプロパティ画面は次のいずれかの手順で開くことができます。

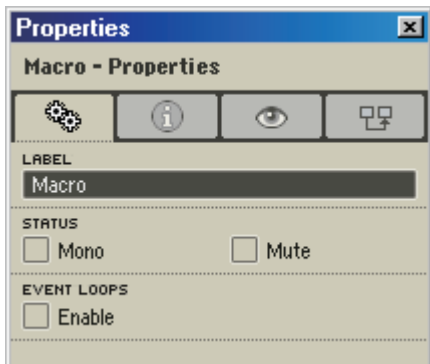
- ・ 灰色い帯上に表示された名前の部分をダブル・クリックする方法。
- ・ マクロ・アイコン上でコンテキスト・メニューを開き、**Properties** コマンドを実行する方法。
- ・ ストラクチャー・ウィンドウでマクロを選択し、**View > Show Properties** コマンドを実行する方法。

プロパティ画面は **Function**、**Info**、**Appearance**、**Connection** の4つのページに分かれており、上部に並んでいる次のボタンでそれぞれ切り替えるようになっています:    .

なお、マクロの場合、**Connection** ページには何もありません。

プロパティ画面は、アンサンプル、インストールメント、基本マクロなどのオブジェクトを選択し直すと、それを反映して表示内容が変わります。したがって、2つのオブジェクトの設定値を見比べたい場合は、画面を開いたまま交互にオブジェクトの選択を切り替えてください。

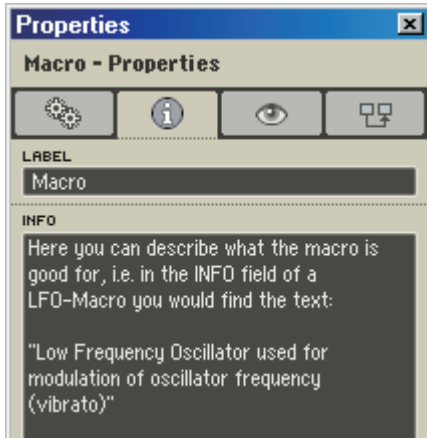
Function ページ




マクロのプロパティ画面、*Function* ページ

- ・ **LABEL:** マクロ名を表し、ここで変更も可能です。ストラクチャー・ウィンドウのマクロ・アイコンにはこの名前が表示されるほか、制御パネル・フレームがある場合はそこにも現れます。
- ・ **STATUS:**
 - ・ **Mono:** オンにすると、マクロ内のモジュールがすべて、単声信号を処理するモードになります。CPU に対する負荷を軽減できるので、多声にする必要がなければオンにしてください。
 - ・ **Mute:** 信号経路上、このマクロの上流に位置するオブジェクトを、すべて無音にします。ストラクチャー・ウィンドウを開くと、状態 LED には赤い「M」印、入出力ポートには赤い「x」印がついています。
- ・ **EVENT LOOPS:** オンにすると、イベント信号のループが生じてもそれを回避しようとしなくなります。但し、処理スタックがあふれ、再生や操作が不能になる恐れがあります。オフにすると、ループが生じそうになったとき、ループの発生源を表示して続行するかどうか訊ねるようになります。REAKTOR の動作を安定させるためにも、通常はオフにしておくようお勧めします。

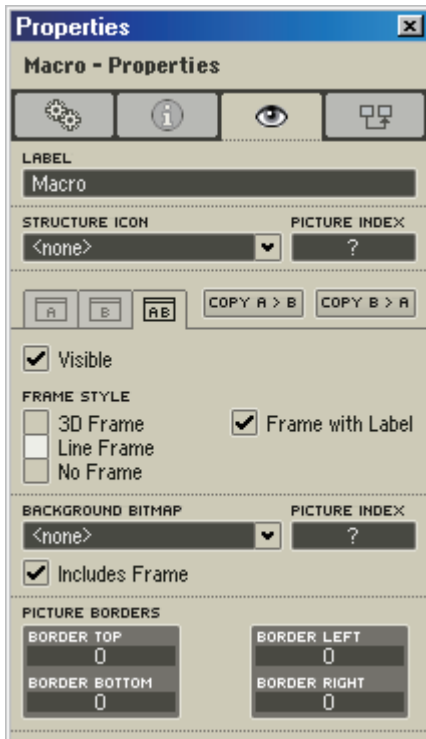
Info ページ



マクロのプロパティ画面、**Info** ページ

- ・ **INFO**: このマクロに関する覚え書きを残しておくことができます。アンサンブル制御パネルやストラクチャー・ウィンドウのツールバーで **Show Info**  をオンにすると、マクロ・アイコンや制御パネル・フレーム上にマウス・カーソルを持っていったとき、この覚え書きが表示されるようになります。

Appearance ページ



マクロのプロパティ画面、*Appearance* ページ

- ・ **STRUCTURE ICON**: ストラクチャー・ウィンドウに表示する際のアイコン画像。特に指定しなければ、3つのモジュールを結線した様子のかたどったアイコンになります。
- ・ **PICTURE INDEX: STRUCTURE ICON** として、複数の画像を縦または横に並べたビットマップを指定した場合、何番目の画像を適用するか、その番号を指定します。なお、画像の個数は **NUM ANIMATIONS** プロパティとして設定します。
- ・ **A、B、AB**: 制御パネルは **A、B** の2つがあって、随時切り替えながら使えるようになっていますが、このマクロをどちらの制御パネルに表示するか、ここで指定します。以下に説明する、**FRAME STYLE**、

PICTURE BORDERS の設定は、ここで指定した制御パネルについてのみ適用されます。

- ・ **COPY A > B, COPY B > A:** 一方の制御パネルの中身を、そのままもう一方にコピーするためのボタンです。
- ・ **FRAME STYLE:** 制御パネル上、このマクロに関係するつまみ類を囲む枠を、次の中から選択できます。
 - ・ **3D Frame:** 浮き上がって見える枠。
 - ・ **Line Frame:** 直線をつないだ枠。
 - ・ **No Frame:** 枠なし。
- ・ **Frame with Label:** オンにすると、マクロ名を枠に添えて表示ようになります。
- ・ **BACKGROUND BITMAP:** 制御パネルの背景に表示する画像を指定します。**A** と **B** で別の画像を設定できます。
- ・ **PICTURE INDEX:** 背景画像として、複数の画像を縦または横に並べたビットマップを指定した場合、何番目の画像を適用するか、その番号を指定します。なお、画像の個数は **NUM ANIMATIONS** プロパティーとして設定します。
- ・ **Includes Frame:** オンにすると、背景画像の周りに枠を描画するようになります。

PICTURE BORDERS: 制御パネルの上下左右に確保する余白を、ピクセル単位で指定します。**A** と **B** でそれぞれ異なる値を指定できます。なお、格子を表示する関係上、4 の倍数で指定してください。

17. 基本ストラクチャー

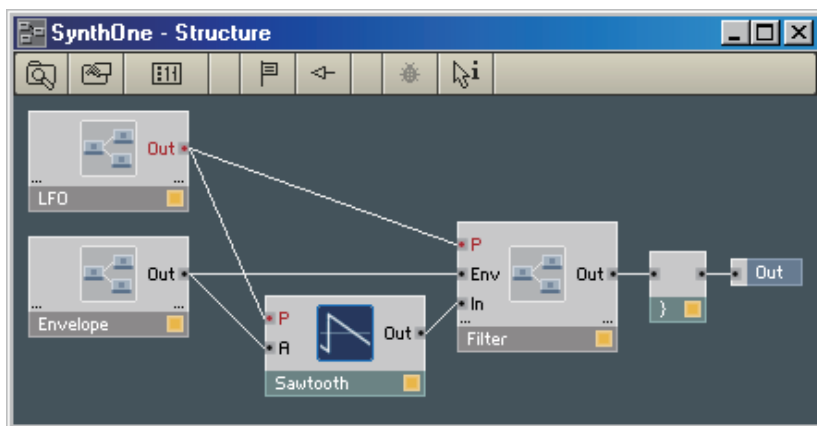
17.1. 「基本ストラクチャー」の役割

REAKTOR は、音作りのためのしかけを思いのままに設計、実現できるよう、内部の部品ひとつひとつを公開して組み合わせ方はユーザーに任せる、という考え方を採用しています。古くからのモジュール型シンセサイザーに通じるところがあるので、各部品のことを「モジュール」と呼びます。なお、REAKTOR 5 では、基本モジュールとコア・モジュールを区別して扱うようになりました。

REAKTOR にはさまざまなモジュールがライブラリーの形で付属しており、これを組み合わせて MIDI 信号やオーディオ信号を処理するようになっています。複雑な信号処理ストラクチャーも、ひとつひとつは単純な機能のモジュールを接続して実現します。

基本ストラクチャー・ウィンドウには、基本モジュールを並べ、相互に接続する作業台の役割があります。

注意：この章では基本ストラクチャー、基本マクロ、基本モジュールのことを、単に「ストラクチャー」、「マクロ」、「モジュール」と呼ぶことにします。「コア・ストラクチャー」、「コア・マクロ」、「コア・モジュール」には当てはまらない説明もあるので注意してください。



ストラクチャー・ウィンドウ

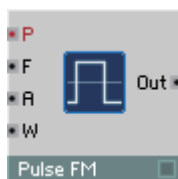
モジュールを組み合わせてストラクチャーを作成する際は、階層化を常に意識するよう心がけてください。「アンサンプル」は最上位階層に位置するストラクチャーですが、この直下には「インストゥルメント」だけを置きます。各「インストゥルメント」は、「マクロ」、「モジュール」、「コア・セル」のみを組み合わせてるものとし、他の「インストゥルメント」を含めることはできるだけ避けてください。また、「マクロ」は他のマクロやモジュール、コア・セルから成り、インストゥルメントを含めることはできません。

複雑なストラクチャーを構築する場合、すっきりと分かりやすい設計をすることが特に重要です。次のような点を心がけてください。

- ・「アンサンプル」の直下には「インストゥルメント」だけを置き、マクロやモジュールは置かないようにしてください。ミキサーのように、いくつかのインストゥルメントの出力信号をミックスするようなものも、ライブラリーには独立したインストゥルメントとして用意されています。
- ・ひとまとまりの機能単位ごとに「マクロ」としてまとめると、見通しがよく管理しやすいストラクチャーになります。発振器、エンベロープ生成器などのように、機能単位が明確であれば、他の用途に再利用するのも容易です。また、問題が起こったときに原因を見つけやすい、という利点もあります。

17.2. モジュール

「モジュール」は最下層に位置する機能単位です。ストラクチャー・ウィンドウ上では、機能を象徴するアイコンを添えた箱の形で表示されます。



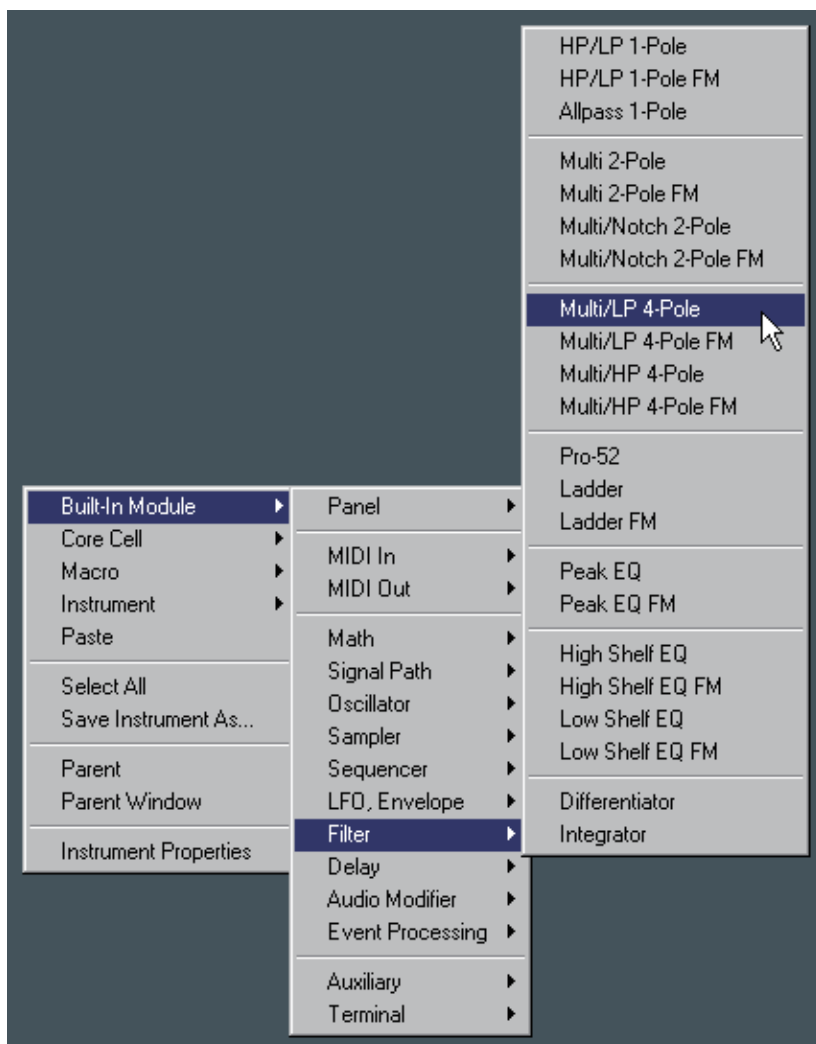
モジュールの例 (Pulse FM)

モジュールをストラクチャーに追加する手順

ストラクチャー・ウィンドウの背景部分でコンテキスト・メニューを開き、**Built-In Module** 以下のサブ・メニューから追加したいモジュールを選択してください。

モジュールは系統別に分かれています。そこで、機能グループ (例えばフィルター系であれば **Filter** メニュー) を開き、その中からモジュール (例えば **Multi/LP 4-Pole**) を選択します。個々のモジュールについては本書の後半、「モジュール・リファレンス」を参照してください。

新しいモジュールはコンテキスト・メニューを開いた位置に追加されますが、どこにでも移動可能です。



新しいモジュールの追加メニュー

モジュールの入出力ポート

各モジュールには「ポート」がいくつかついており、ここを介して他のモジュールと接続します。左辺には入力ポート、右辺には出力ポートがあります。

入力ポートに何もつながず開放にしておくと、値が 0 の信号があるように振る舞います。つまり、値が 0 の定数信号源をつないだのと同じことです。

ポートを介してやり取りする信号には、「オーディオ信号」と「イベント信号」の 2 種類があります。

- ・オーディオ信号は主として音声を運ぶもので、アナログ処理では電圧変化により表していたものに相当します。オーディオ信号処理は、CPU に対して相応の負荷がかかります。オーディオ信号をやり取りするポートは、ラベルが黒い文字になっています。オーディオ入力ポートに複数の信号を接続することはできません。必要に応じ、**Add**、**Amp/Mixer** などミキサー系のモジュールで、1 本の信号にまとめてから接続してください。既に結線がつながっている入力ポートに重ねて接続しようとする、元からあった結線は削除されてしまいます。
- ・イベント信号は、あるパラメーター値が変わったという制御メッセージを運ぶものです。これを扱う代表的なモジュールとしては、MIDI 入力系や、フェーダーなどパネル系のモジュールがあります。複雑な演算処理は断続的にしか必要としないので、CPU に対する負荷はそれ程高くありません。イベント信号をやり取りするポートは、ラベルが赤い文字で、赤丸がついています。イベント入力ポートに複数の信号を接続する場合は、**Merge** モジュールで 1 本の信号にまとめてください。「ゲート信号」はイベント信号の中でも特殊なもので、値が正であればゲートを開き、0 または負であれば閉じることを表します。オーディオ信号とイベント信号の両方に対応したモジュールもあります。初めはイベント・モジュールになっていますが、入力ポートにオーディオ信号を接続すればオーディオ・モジュールに変わります。接続信号数を増やすと CPU への負荷も高くなります。

ポート上でコンテキスト・メニューを開くと、次のようなコマンドが並んでいます。

- ・ **Create Control**: ポートの種類に応じたつまみ類を制御パネル上に追加します。詳しくは第 18 章「制御パネルの編集」、第 19 章「パネル操作」を参照してください。

- ・ **Create Constant:** 適当な値の定数信号源を生成、接続します。
- ・ **Mute Port:** 無音、すなわち値を 0 にします。ポートには赤い「x」印が付きます。

モジュールに関するコンテキスト・メニュー

- ・ **Mono:** オフにすると、多声信号、すなわちいくつかの声部の信号を並列に処理するモードになります。処理できる声部の最大数は、モジュールが属するインストゥルメントの設定によって決まります。多声モードのモジュールは、左下隅の状態 LED が黄色になります。一方、オンにすると、単声信号、すなわち声部数が 1 の信号を処理するモード、状態 LED は橙色になります。なお、プロパティー画面で切り替えることも可能です。CPU に対する負荷を軽減できるので、多声にする必要がなければオンにしてください。
- ・ **Mute:** オンにすると、信号処理を行わず、何も出力しない（無音）ようになります。プロパティー画面で切り替えることも可能です。無音状態のモジュールは、状態 LED に赤い「x」印が付きます。CPU に対する負荷はほとんどありません。一時的にモジュールの機能を無効にしたいだけならば **Mute** をオンにし、将来にわたって不要であればモジュール自体を削除するとよいでしょう。出力ポートに何もつながっていない、あるいは無音状態のモジュールしかつながっていない場合、自動的に無音状態になり、状態 LED も消灯します。したがって、信号の行き先を切り替えるスイッチと組み合わせれば、使っていない側のモジュールを無音状態にすることにより、CPU に対する負荷を軽減できます。
- ・ **Cut:** モジュールをクリップボードに複製し、ストラクチャー上から削除します。これを **Paste** コマンドで別の位置に貼り付けることができます。
- ・ **Copy:** 同様にクリップボードに複製しますが、ストラクチャー上からは削除しません。
- ・ **Duplicate:** モジュールを複製し、同じストラクチャー上に貼り付けます。**Copy**、**Paste** の 2 つのコマンドを実行したのと同事です。
- ・ **Delete:** モジュールをストラクチャー上から削除します。
- ・ **Properties:** モジュールのプロパティー画面を開きます。個々のモジュールについては本書の後半、「モジュール・リファレンス」を参照してください。

17.3. 信号源モジュール

概要

「信号源モジュール」とは制御信号を出力するモジュールのことで、次の3種類があります。

- ・ **制御信号源**：制御パネル上にフェーダーやノブの形で表示され、これを使用して値を設定できます。
- ・ **MIDI 信号源**：MIDI データを制御信号に変換します。
- ・ **定数信号源**：固定値を出力します。

制御信号源

制御信号源としては例えば、**Fader**、**Knob**、**Button** などのパネル系モジュールがあります。これをストラクチャーに追加する手順が2通りあります。

- ・ ストラクチャー・ウィンドウ上で、コンテキスト・メニューから選択、追加する方法。
- ・ モジュールの入力ポート上でコンテキスト・メニューを開き、**Create Control** コマンドを実行する方法。入力ポートに接続するところまで自動で進むので、種類、ラベルなどを適宜変更してください。こちらの方法の方が、手間がかからないことが多いでしょう。

制御信号源やこれに対応する制御パネル上のつまみ類は、MIDI 信号でも制御できます。

制御信号源上でコンテキスト・メニューを開くと、**MIDI Learn**、**Cut**、**Copy**、**Duplicate**、**Delete**、**Properties** といったコマンドが並んでいます。

MIDI 信号源

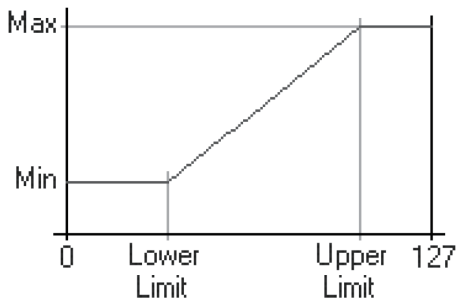
MIDI イベントを使ってオーディオ信号処理を制御するために使います。MIDI イベントの種類ごとに専用のモジュールがあり、該当するMIDI イベントにのみ反応して信号を出力します。例えば **On Velocity** モジュールは、MIDI キーボードを操作したとき、MIDI ノート・オン・メッセージを受けてベロシティ情報を出力します。

コンテキスト・メニュー上では、**Built-In Module > MIDI In** 以下に並んでいます。

値の範囲

制御信号源や MIDI 信号源は、出力信号の値の範囲を、**Min**、**Max** プロパティで設定できます。これはプロパティ画面の **Function** ページにあります。

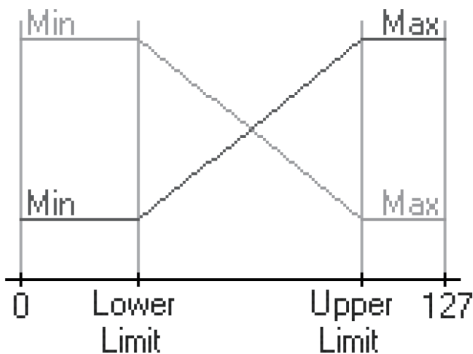
また、MIDI 信号源の場合、入力信号として受け付ける値の範囲を、**Lower Limit**、**Upper Limit** プロパティで設定できます。この範囲の入力値が、**Min** ～ **Max** の範囲に、線形に写像されることになります。



入力信号と出力信号の対応

Lower Limit、**Upper Limit** の値は 0 ～ 127 の範囲で、しかも **Lower Limit** の方が小さな値でなければなりません。

しかし **Min** と **Max** は、大小が反転していても構いません。正反 2 種類の信号源を 2 つ組み合わせることにより、クロスフェードを実現できます。



クロスフェード

ある値(閾値)より上か下かに応じて2種類の値を出力する「スイッチ」は、**Lower Limit** にその閾値、**Upper Limit** にこれより1大きい値を指定する(例えば63と64)ことで実現できます。この場合、入力値が63以下ならば**Min**、64以上ならば**Max**の値が出力されることになります。

刻み幅

出力値の刻み幅(分解能)は通常128ですが、**Fader**、**Knob**などのモジュールでは、**Stepsize**としてこれより小さな値を設定できます。例えば1オクターブ単位でピッチを調整したい場合、**Stepsize**を12としてください。

定数信号源

固定値を出力するモジュールやマクロのことです。具体的な値は**Function**ページの**Value**プロパティで指定します。

コンテキスト・メニューの**Built-In > Math > Constant**で追加してください。

17.4. スイッチ

これは信号源ではありませんが、制御パネル上に切り替えスイッチの形で表示されるのでここで説明します。

入力ポートにそれぞれモジュールやマクロを接続しておき、どの入力信号を通すか、スイッチ操作で切り替えます。但し**Toggle**モードの場合は例外で、1系統の信号のオン/オフを切り替えるようになっています。詳しくは「モジュール・リファレンス」を参照してください。

スイッチを積極的に活用すると、CPUに対する負荷を大幅に軽減する効果があります。オフになっている入力ポートにはオーディオ信号を供給する必要がないため、この経路上のモジュールも自動的にオフになるからです。例えば数種類の発振器をつないでおき、スイッチ操作によりいずれか1つの発振信号のみを出力する場合、それ以外の発振器モジュールはオフになります。

17.5. 入出力端子

ストラクチャー上あまり目立ちませんが、非常に重要なモジュールです。ハードウェアでいう「ソケット」や「プラグ」に相当します。スト

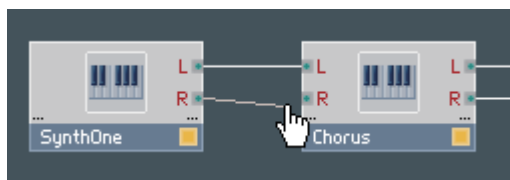
ラクチャーの上位階層で見ると「ポート」の形で現れ、他のインストールメントやマクロ、モジュールとの接続に使います。

ポートの種類に応じて、**In Port**、**Out Port**、**Send**、**Receive**、**IC Send**、**IC Receive**、**OSC Send**、**OSC Receive** の 8 種類があります。結線についても他のモジュールと同様です。

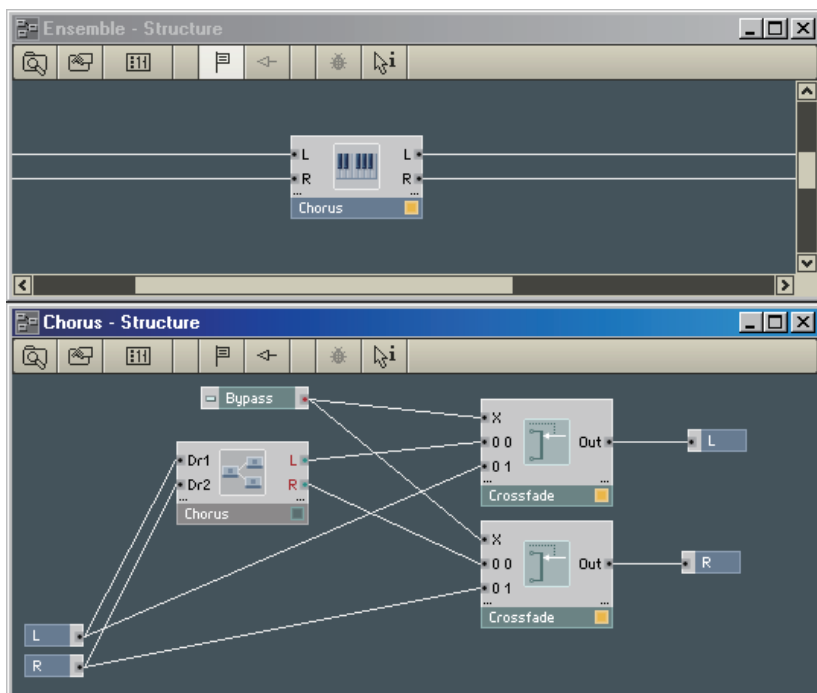
入出力端子系モジュールは、コンテキスト・メニューの **Built-In Module > Terminal...** 以下に並んでいます。ラベルは単に「In」、「Out」となっていますが、複数のポートを設ける場合など、必要に応じて(次の図の **L**、**R** のように)変更しても構いません。また、説明文をプロパティ画面の **Info** ページに書いておくともよいでしょう。上位階層ストラクチャーで見ると、ラベルがポートに添える形で表示されるほか、マウス・カーソルをポート上に持っていくと説明文が現れます。

17.6. 結線

モジュールやマクロのポート間を結ぶ線を「結線」と言います。信号はこの結線上を伝播します。



結線による接続



4つのポートがあるインストゥルメントと、

対応する入出力端子が内部ストラクチャーに使われている様子

結線による接続

- ・ モジュールやマクロの出力ポート上にマウス・カーソルを持っていき、左ボタンを押して（クリックして）そのままドラッグしてください。接続先の入力ポート上で放すと、結線で接続されます。

切断

結線を削除（切断）する方法は2通りあります。

- ・ 接続先の入力ポート上をマウスでつかんでドラッグし、何もない領域で放す方法。
- ・ 結線上をクリックして選択し、**delete** キーを押す方法。

結線に関する規則

結線に関しては次の規則があります。

- ・ 接続できるのは、出力ポートから入力ポートに向かう結線に限ります。入力ポートどうし、出力ポートどうしは接続できません。
- ・ ある出力ポートから出る結線は最大 40 本までです。
- ・ 入力ポートを開放にしておくと、値が 0 の信号が供給されているように振る舞います。
さらに次のような規則もあります。
- ・ イベント入力ポートでは、オーディオ信号を処理できません。必要に応じ、**A to E** モジュールでオーディオ信号をイベント信号に変換してください。
- ・ イベント出力ポートからは、イベント入力だけでなくオーディオ入力ポートにも接続できます。
- ・ 単声信号を多声入力ポートに供給すると、各声部とも同じ値の信号が届いたものとして振る舞うようになります。例えばピッチを表す単声信号を与えると、斉奏 (ユニゾン) と同じことになります。
- ・ 多声信号を単声入力ポートに供給することはできません。この場合、入力ポートに赤い「x」印が現れます。必要に応じ、**Audio Voice Combiner** モジュールで変換してください。

信号経路上の値の表示

Show Info をオンにすると、結線上にマウス・カーソルを持っていったとき、信号の値が表示されるようになります。

イベント信号の場合は、直近にこの経路上を流れたイベントの値です。表示を更新する頻度よりもイベント信号の頻度が高ければ、表示されないでしまう値が出てきます。

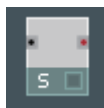
オーディオ信号の場合は信号の最小値と最大値のみ表示します。但し、一瞬だけ値が大きくなっても、表示に反映されない場合があります。値の範囲が常に変動する信号の場合、マウス・カーソルを放していったん値表示をクリアし、改めてマウス・カーソルを持っていく必要があるかも知れません。

多声信号の場合は、声部ごとに並べて表示されます。各声部の欄の左側は MIDI ノート番号です。ない場合は一段低い位置に、「Note: Off」という表示が現れます。

17.7. 信号処理

REAKTOR では、イベント信号とオーディオ信号の 2 種類を区別して扱います。イベント信号は毎秒数百回程度のレートで処理しますが、オーディオ信号はサンプル・レートに相当する、毎秒数万回の処理量になります。例えばコンパクト・ディスクの標準的なオーディオ信号ならば毎秒 44100 回で、通常はこれを 44.1kHz と表します。このように 2 通りのレートを区別して処理することにより、CPU への負荷を軽減しています。具体的なレート値は **Settings** メニューで調整できます。

オーディオ生成 / 処理系のモジュールは、もちろんオーディオ・レートで処理します。イベント信号をイベント・レートで出力するモジュールとしては、**Event Smoother**、**LFO**、**Slow Random**、**A to E** などがあります。



Event Smoother モジュール



LFO モジュール



Slow Random モジュール



A to E モジュール

イベント・モジュールの中には、一定頻度でイベントを監視するのではなく、新しいイベントが届いたときのみ処理するものもあります。具体的には、パネル上のつまみ類をマウスで調整したときに発生するマウス・イベントや、MIDI イベント、場合によってはオーディオ・イベントがこれにあたります。**A to E Trig** モジュールのようにオーディオ信号を常時処理する場合は、オーディオ・レートと同じ頻度でイベント信号を出力します。

イベント入力ポートでは、レートにかかわらず、入ってくるイベントをすべて処理します。例外として **Iteration** モジュールは、所定のオーディオ・レートでイベントを処理します。最後に、状況によってどちらのレートでもイベントを処理できる、混成モジュールというものもあります。例えば数学演算系モジュールがこれにあたります。次のように、ポートに添えられた点の色で、実際のレートがわかるようになっています。

- ・ 混成モジュールのポート上に緑の点があれば、まだ接続されておらず、イベント信号でもオーディオ信号でもつなげる状態です。
- ・ 赤の点があればイベント・ケーブルがつながっており、イベント・レートで処理することを表します。
- ・ 黒の点があればオーディオ・ケーブルがつながっており、オーディオ・レートで処理することを表します。

イベント信号

イベント信号は、あるパラメーター値が変わったという制御メッセージを運ぶものです。これを扱う代表的なモジュールとしては、MIDI 入力系や、フェーダーなどパネル系のモジュールがあります。複雑な演算処理は断続的にしか必要としないので、CPU に対する負荷はそれ程高くありません。イベント信号をやり取りするポートは、ラベルが赤い文字で、赤丸がついています。イベント入力ポートに複数の信号を接続する場合は、**Merge** モジュールで 1 本の信号にまとめてください。オーディオ

の出力ポートからイベント入力ポートへは、**A to E** の変換モジュールを使って接続できます。

「ゲート信号」はイベント信号の中でも特殊なもので、値が正であればゲートを開き、0 または負であれば閉じることを表します。

各イベントはその発生時刻と値の組で区別されます。

イベント信号はオーディオ信号として扱うこともできますが、レートが低いので、次のイベントが来るまでの間は一定の値であるとみなします。つまり、連続的にではなく、階段状に変化する信号ということです。

A to E、**LFO** のように、オーディオ信号を例えば 200 回 / 秒のイベント・レートで処理するモジュールも若干あります。しかしほとんどの場合、イベントが届いた時点でその都度処理するようになっています。

イベント処理の順序

ほとんどのイベント処理モジュールは、イベント入力に応じて直ちに出力イベントを生成します。したがって、信号経路に沿ってイベントが高速で伝播することになります。

複数の結線が出ている出力ポートがあれば、どの経路にもイベントが伝播します。このときの伝播順序は「深さ優先」で決まります。すなわち、一方の経路を伝わって出口点まで達した後、元に戻って次の経路に移ります。

これに当てはまらない順序で処理したい場合は、**Order** モジュールを使ってください。

もうひとつ、**Value** モジュールについても解説しておきましょう。**Val** 入力に値を与えますが、その時点ではイベントを出力しません。トリガー入力にイベントが与えられた時点で出力します。イベント制御型のサンプル & ホールド機能と考えてもよいでしょう。**Order** モジュールでトリガー・イベントを作ることにより、処理順序を自由に制御できます。

いくつかの信号源モジュールが同時にイベントを生成する場合（初期設定の際など）、その実際の順序は、ストラクチャーにモジュールを追加した順序によって決まります。モジュールを **Cut** し、そのまますぐに **Paste** することにより、この順序を変えることができます。

イベント・ループの回避

環境設定の **Options** ページで **Globally disable event loops** をオンにする、あるいはアンサンプル / インストゥルメント / マクロの **Event Loops Enable** プロパティをオフにする (**Function** ページ) により、イベント・ループを回避することができます。ループが生じると、処理スタックがあふれ、再生不能、あるいは操作不能に陥る可能性があります。

注意：ループが生じると REAKTOR が操作不能になってしまうのは、バグではなく仕様です。ストラクチャー設計の際に気を配るしかありません。**Value** モジュールを適切に組み込むことにより、安全性を高めることができます。

上記のように設定すると、イベント信号のループが生じそうになったとき、ループの発生源を表示して続行するかどうか訊ねるようになります。

ループが生じると、処理スタックがあふれ、再生不能、あるいは操作不能に陥る可能性があります。その場合、いったん REAKTOR を終了して再起動し、**Globally disable event loops** をオンにして動かして、表示されるループ発生源を手掛かりに問題点を特定してください。このときオーディオ出力もオフにすると、ループが何重にも重なって制御しきれなくなるのをある程度防ぐことができます。

REAKTOR の動作を安定させるためにも、通常はオンにしておくようお勧めします。なお、旧版で保存したファイルを読み込んだ場合は、互換性を保つためオフになっています。

注意：多くの場合、**Iteration** モジュールを使うことにより、イベント・ループを使わなくても同等の処理ができます。このモジュールには処理速度を制限するプロパティがあるので、ループが速過ぎる場合に生じる雑音も回避できます。

オーディオ信号

オーディオ信号は主として音声を運ぶもので、アナログ処理では電圧変化により表していたものに相当します。オーディオ信号処理は、CPU に対して相応の負荷がかかります。オーディオ信号をやり取りするポー

トは、ラベルが黒い文字になっています。オーディオ入力ポートに複数の信号を接続することはできません。必要に応じ、**Add, Amp/Mixer** などミキサー系のモジュールで、1本の信号にまとめてから接続してください。既に結線がつながっている入力ポートに重ねて接続しようとすると、元からあった結線は削除されてしまいます。

オーディオ・モジュールの有効 / 無効の切り替え

オーディオ・モジュールはCPUに対する負荷が高いため、その出力が最終的に **Audio Out** モジュールにつながっているのであれば、自動的に無効にするようになっています。有効なモジュールは、右下のLEDが点灯しています。

但しこれには例外があって、ランプ系などはどんな場合であっても有効になるよう、プロパティで設定するのが普通です。これに信号を供給するモジュールもやはり、常に有効になります。ランプ自身はオーディオ信号を出力しませんが、上流につながっているモジュールが有効かどうか表示するのが役割なので、自動的に無効になってしまっは困ります。

オーディオ処理の順序

イベント信号の処理順序は、ストラクチャーにモジュールを追加した順序によっても左右されますが、オーディオ信号はこれとは違い、信号の流れに沿って処理されます。実際の処理順序は、**System > Debug > Show Module Sorting** コマンドで確認できます。

この方式は分かりやすいのですが、フィードバック・ループがある場合に問題が生じます。しかしループが役に立つ場合もあるので、一律には禁止せず、処理順序を適宜調整することにより回避するしくみになっています。ループの先頭モジュールは、該当するポートに青い縦線が入っていますが、これはループ回避のために **Unit Delay** モジュールを自動的に挿入したことを表します。このモジュールは表には現れませんが、積極的に **Unit Delay** モジュールを挿入することにより、別のモジュールがループの先頭であるように扱うことも可能です。

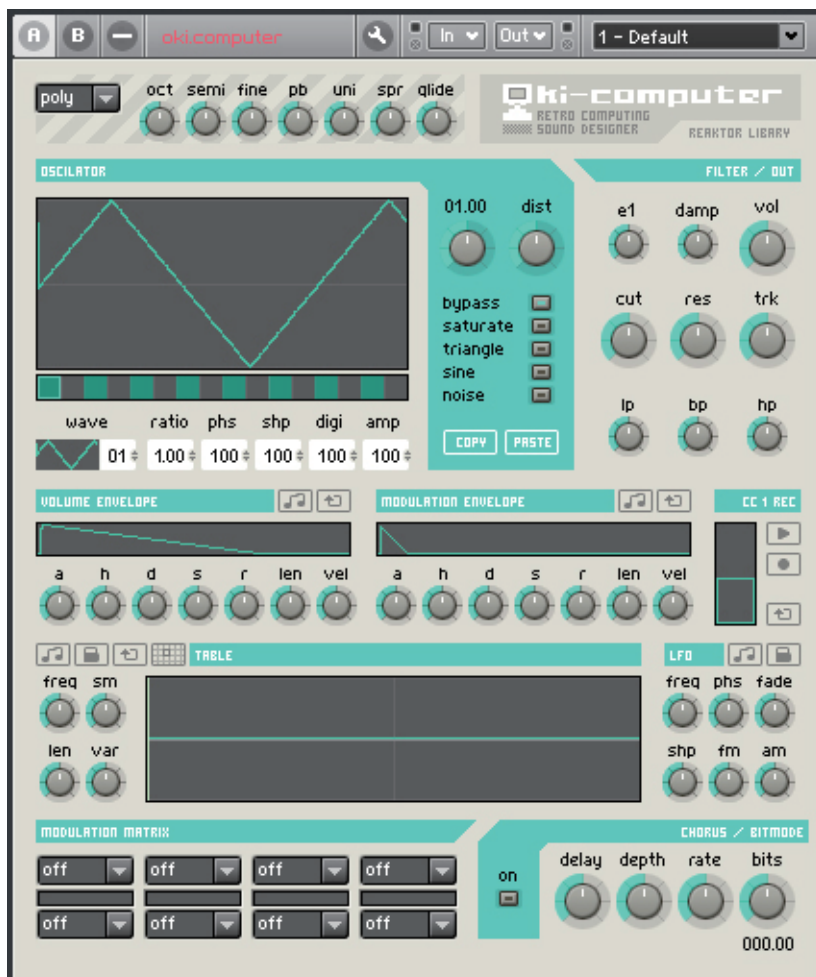
17.8. コンテキスト・メニュー

ストラクチャー・ウィンドウ上でコンテキスト・メニューを開くと、次のようなコマンドが並んでいます。

- ・ **Built-In Module:** モジュールを追加。サブ・メニュー以下でモジュールを選択してください。
- ・ **Core Cell:** コア・セルを追加。
- ・ **Macro:** マクロを追加。
- ・ **Instrument:** インストゥルメントを追加。
- ・ **Paste:** あらかじめクリップボードに複製しておいた内容を貼り付け。貼り付け位置を指定してから実行してください。
- ・ **Select All:** ストラクチャー上のオブジェクトをすべて選択状態にします。
- ・ **Save Instrument/Macro As...:** ストラクチャーに新しい名前をつけてファイルに保存します。インストゥルメントならば *.ism、マクロならば *.mdl という拡張子になります。
- ・ **Parent:** 上位階層のストラクチャー表示に切り替えます。例えば、あるインストゥルメントに属するマクロを編集集中であれば、**Parent** を実行するとインストゥルメントの編集画面に切り替わります。
- ・ **Parent Window:** 別のウィンドウを開き、上位階層のストラクチャーを表示します。
- ・ **Instrument/Macro Properties:** プロパティ画面を開きます。

18. 制御パネルの編集

18.1. 制御パネル



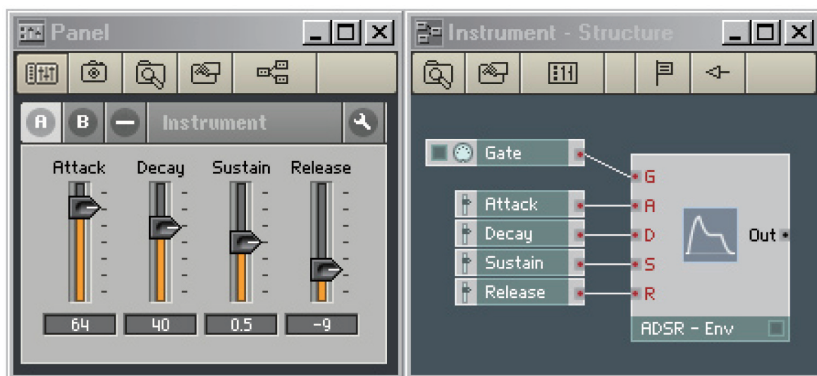
制御パネルの例。REAKTOR 5 に付属の「OKI Computer」アンサンブル

「制御パネル」とは、インストゥルメントの動作設定に使うユーザー・インターフェイスです。ハードウェアで実現されたシンセサイザーやエ

フェクターのフロント・パネルと同様に、フェーダーやノブ、スイッチが並んでいます。各インストゥルメント用のパネルが、アンサンプル用のパネル内に並ぶ形で表示されます。

18.2. 制御パネルのつまみ類

発振器、フィルター、サンプラーなど、オーディオ信号を生成 / 加工するモジュールのほかに、ノブ、フェーダー、ボタンなどの「つまみ類」を操作して値を変えることにより、信号の流れを制御するモジュールがあります。



左：フェーダーが並んだ制御パネル、右：これに対応するストラクチャー

18.3. 各種のつまみ類

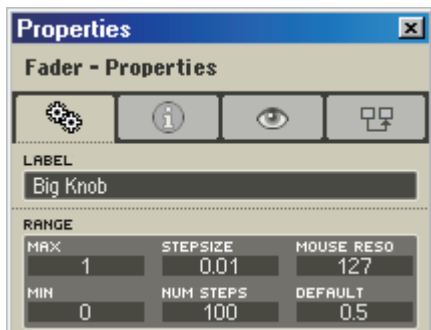
REAKTOR でよく使われるつまみ類として、フェーダー、ノブ、ボタン、スイッチ、リストについて順に解説します。

フェーダー、ノブ



各種のフェーダー、ノブ

上下 / 左右に動かす、あるいは回すことにより値を調節し、その値を他のモジュールに与えるようになっています。例えば、発振器の **P** 入力に与えて発振周波数 (ピッチ) を変える、サンプラーの **A** 入力に与えて振幅を調整する、などという使い方があります。出力値の範囲は **Min**、**Max**、刻み幅は **Stepsize** の各プロパティーで設定できます。また、マウスの移動量に対して値が変化する割合は **Mouse Reso** として設定します。



プロパティー画面、*Function* ページ

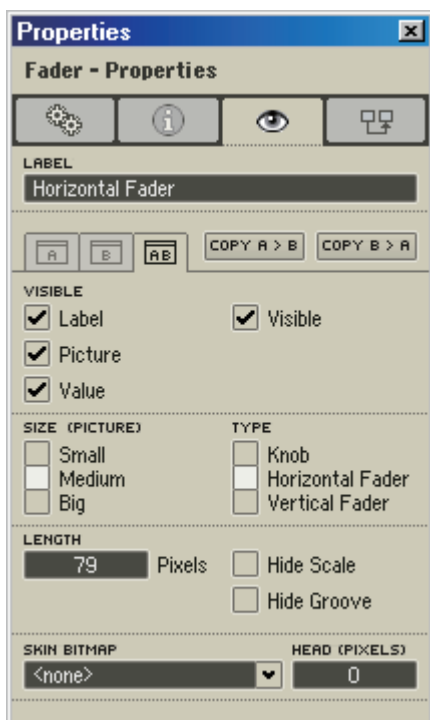
Stepsize を 0 にすると、**Min** ~ **Max** の範囲を 127 段階に刻む分解能になります。多く場合これで充分でしょう。

値の調節は、マウスでつかんでドラッグする方法のほか、上下の矢印キーでも可能です。この場合、フェーダー / ノブを選択しておいてから、

矢印キーで調節してください。さらに、MIDI 信号で調節することもできます。

ノブの場合、マウスを左右ではなく上下に動かして調節します。

制御パネル上の表示形態は、プロパティー画面の **Appearance** ページで設定します。



プロパティー画面、**Appearance** ページ

- ・ **VISIBLE:** (Label、Picture、Value、Visible) ラベル、ノブやフェーダーの画像、値、全体のそれぞれについて、表示 / 非表示を切り替えます。
- ・ **SIZE:** (Small、Medium、Big) 表示の大きさを切り替えます。
- ・ **TYPE:** (Knob、Horizontal Fader、Vertical Fader) ノブ、横方向のフェーダー、縦方向のフェーダーの中から選択します。**Fader** モ

ジュールを使い、制御パネル上ではノブとして表示する、といったことも可能です。

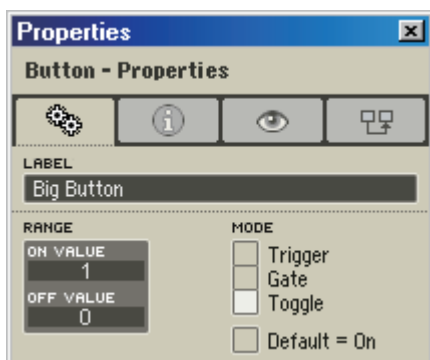
- ・ **LENGTH**: フェーダーの長さ (縦方向の場合は高さ) をピクセル単位で表します。ノブの場合は指定を変えても何の効果もありません。
- ・ **Hide Scale**、**Hide Groove**: 目盛、つまみが動く溝の表示 / 非表示を切り替えます (溝はフェーダーのみ)。
- ・ **SKIN BITMAP**、**HEAD (PIXELS)**: 18.4 節「つまみ類のスキン」を参照。

ボタン



各種のボタン

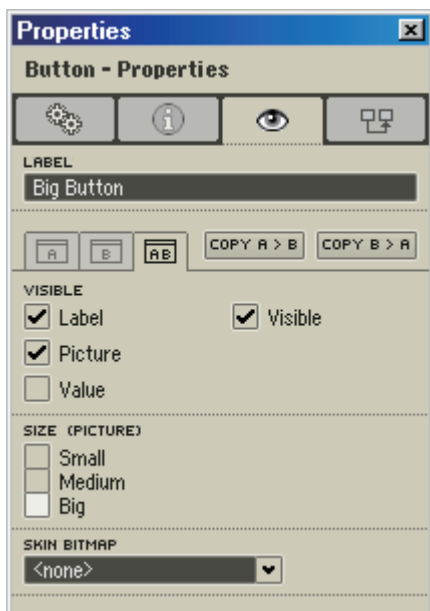
オン / オフにより 2 つの状態を切り替え、それぞれに応じた値を他のモジュールに与えるようになっています。サンプラーのゲート入力につないでトリガーとして使う、などという使い方があります。値は **On Value**、**Off Value** プロパティーで設定できます。



プロパティー画面、*Function* ページ

マウス・ボタンを押すことにより状態が切り替わります。さらに、MIDI 信号で調節することも可能です。

制御パネル上の表示形態は、プロパティ画面の **Appearance** ページで設定します。



プロパティ画面、*Appearance* ページ

- ・ **VISIBLE: (Label, Picture, Value, Visible)** ラベル、ボタンの画像、値、全体のそれぞれについて、表示 / 非表示を切り替えます。
- ・ **SIZE: (Small, Medium, Big)** 表示の大きさを切り替えます。
- ・ **SKIN BITMAP:** 18.4 節「つまみ類のスキン」を参照。

スイッチ



各種のスイッチ

いくつかの入力信号の中から 1 つを選んで出力ポートに送るモジュールです。例えば鋸波と正弦波の発振器を 2 つの入力ポートにつないでおき、スイッチ操作でどちらの発振信号を使うか切り替える、といった使い方があります。

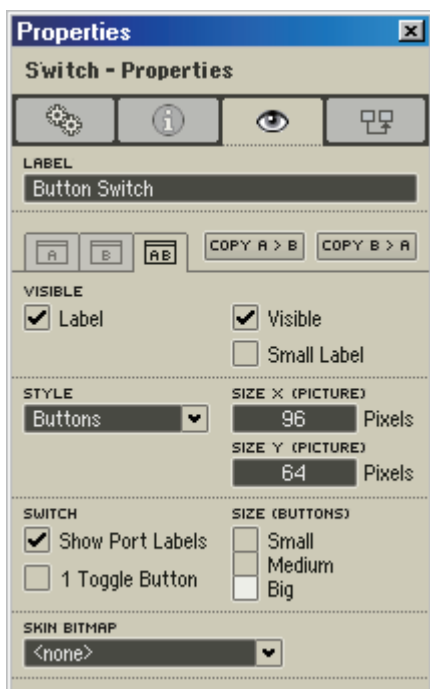


Switch モジュール

このように複数の入力信号から 1 つを選ぶ使い方のほか、入力信号は 1 つだけで、そのオン / オフを切り替える、という使い方も可能です。

マウス操作のほか、MIDI メッセージで切り替えることも可能です。

制御パネル上の表示形態は、プロパティー画面の **Appearance** ページで設定します。



プロパティー画面、*Appearance* ページ

- ・ **VISIBLE: (Label, Picture, Value, Visible)** ラベル、スイッチの画像、値、全体のそれぞれについて、表示 / 非表示を切り替えます。
- ・ **STYLE: (Buttons, Menu, Text Panel, Spin)** 切り替え方法を、ボタン、メニュー、テキスト・パネル、スピンの中から選択します。「テキスト・パネル」はメニューの一形態ですが、選択肢が初めからすべて見えています。「スピン」もメニューの一種で、**+/-** キーで切り替えるようになっています。
- ・ **SIZE X、SIZE Y:** 高さと幅をピクセル単位で指定します。ボタン以外の場合に有効です。
- ・ **SWITCH: Show Port Labels** はボタンにラベルを表示するか否かを表します。**1 Toggle Button** をオンにすると、先頭の入力ポートに対応するボタンしか表示されなくなります (下記の注を参照)。
- ・ **SIZE: (Small, Medium, Big)** 表示の大きさを切り替えます。
- ・ **SKIN BITMAP、HEAD (PIXELS):** 18.4 節「つまみ類のスキン」を参照。

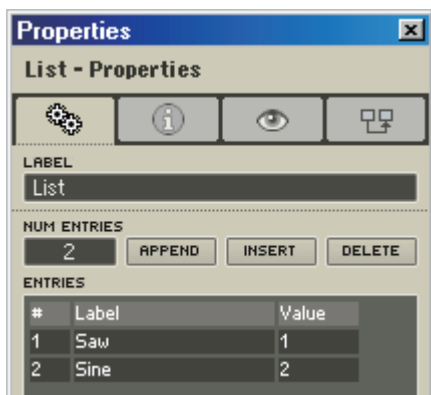
オン / オフなど 2 つの状態を切り替える場合は、**1 Toggle Button** をオンにし、ボタンを 1 つだけ表示するという方法も使えます。

リスト



各種のリスト

いくつかの項目の中から 1 つを選び、これに対応する値を出力ポートに送るモジュールです。項目名と値の対応は、プロパティー画面の **Function** ページ、**ENTRIES** 欄に設定します。

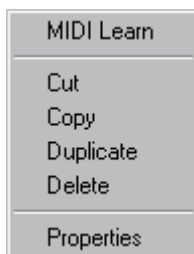


プロパティー画面、**Function** ページ

- **NUM ENTRIES**: リストに並べる項目数を表します。
- **APPEND**: 下記の **ENTRIES** の末尾に項目を追加します。
- **INSERT**: 選択した項目の直下に新しい項目を追加します。
- **DELETE**: **ENTRIES** から項目を削除します。
- **ENTRIES**: (**#**、**Label**、**Value**) 項目の番号、ラベル、値を表します。
マウス操作のほか、MIDI メッセージで切り替えることも可能です。
制御パネル上の表示形態は、プロパティー画面の **Appearance** ページで設定します。その内容はスイッチの場合と同様です。

コンテキスト・メニュー

以上のつまみ類の上でコンテキスト・メニューを開くと、次のようなコマンドが並んでいます。



フェーダー、ノブ、ボタン、スイッチ、リストに関するコンテキスト・メニュー

- ・ **MIDI Learn:** 外部の MIDI コントローラーとの対応、例えばあるノブを MIDI キーボードで制御する、といった関係を設定します。アンサンプル・ツールバーにも同様の機能のボタンがあるので参照してください。
- ・ **Set to Default:** (プロパティーとして設定された) デフォルト値に戻します。
- ・ **Show in Structure:** 対応するモジュールがあるストラクチャーのウィンドウを開きます。
- ・ **Properties:** プロパティー画面を開きます。

18.4. つまみ類のスキン

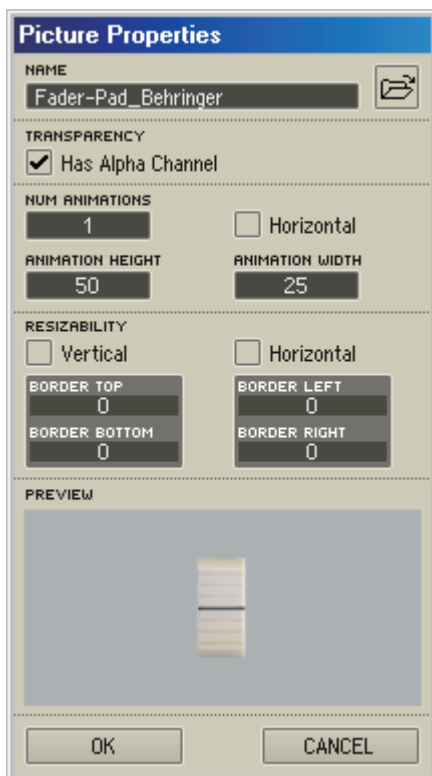
つまみ類の外観を、「スキン」の形でさまざまに変えることができます。前節で解説したつまみ類のほか、「信号受信」モジュール (**Receive**)、ランプ、メーターにもこの機能があります。

フェーダーのスキン

1 枚の画像を使う方式と、複数の画像を使う方式があります。

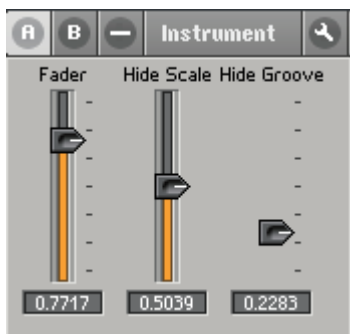
1 枚の画像の場合、この画像は、マウスでつかんで動かす部分 (ハンドル) の表示に使います。 **Picture Properties** 画面で **RESIZABILITY** をオンにすると、フェーダーの大きさに合わせて適当に拡大 / 縮小するようになります。

フェーダーの **HEAD (PIXELS)** プロパティー (**Appearance** ページ) に正の値を指定すると、ハンドル上に、そのピクセル幅のヘッド (目盛を正確に読み取るための矢線) が表示されるようになります。



フェーダーの **Picture Properties** 画面。1 枚の画像を使う場合

一方、複数の画像を使う方式の場合は、ハンドルだけでなくフェーダー全体の外観を変えてしまうことができます。フェーダーの大きさはこの画像の大きさで決まり、**LENGTH** プロパティは無視されます。同様に、**Picture Properties** 画面の **RESIZABILITY** プロパティも無効です。フェーダーの値を何段階に調節できるかは、画像の枚数によって決まります。



目盛や溝を非表示にしたフェーダー

いずれの場合でも、**Hide Scale**、**Hide Groove** プロパティーで、目盛、つまみが動く溝の表示 / 非表示を切り替えることができます。



フェーダーのプロパティー画面、*Appearance* ページ

ノブのスキン

ノブの場合は、複数の画像を使って動きを表す形のスキンしか使えません。ノブの大きさはこの画像の大きさで決まり、**LENGTH** プロパティーは無視されます。同様に、**Picture Properties** 画面の **RESIZABILITY** プロパティーも無効です。ノブの値を何段階に調節できるかは、画像の枚数によって決まります。



ノブのプロパティー画面



さまざまなスキンを適用したノブ

ボタン (およびリスト、スイッチ、信号受信) のスキン

ボタンの場合、4つの画像で「Off state Up」、「On state Up」、「Off state Down」、「On state Down」の4つの状態を表します。**Picture**

Properties 画面で **RESIZABILITY** プロパティーをオンにすると、必要に応じて拡大 / 縮小して表示ようになります。



ボタンの **Picture Properties** 画面。4 つの画像を指定した様子

リスト、スイッチ、信号受信の各モジュールは、**STYLE** プロパティー (**Appearance** ページ) を **Button** とした場合、同様にして外観を変えることができます。



ボタン型リスト / スイッチの外観を変えた様子

ランプのスキン

2つの画像で「Off state」、「On state」の2つの状態を表します。**Picture Properties**画面で **RESIZABILITY** プロパティをオンにすると、**Size X**、**Size Y** プロパティに合わせて拡大 / 縮小して表示するようになります。

メーターのスキン

「オン / オフ」型と「アニメーション」型の、2種類のスキンがあります。

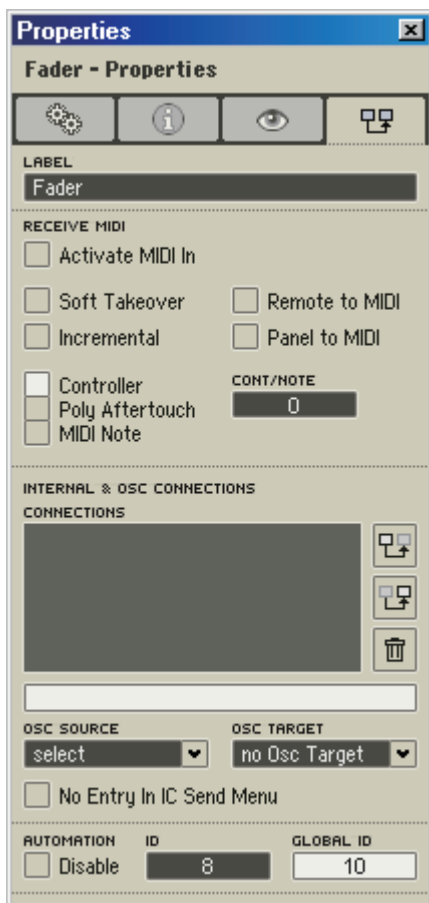
「オン / オフ」型の場合は、2つの画像で「Off state」、「On state」の2つの状態を表します。**Picture Properties**画面で **RESIZABILITY** プロパティをオンにすると、**Size X**、**Size Y** プロパティに合わせて拡大 / 縮小して表示するようになります。

「アニメーション」型の場合は、一連の状態に応じて複数の画像を使います。メーターの表示を何段階に刻むかは、画像の枚数によって決まり、**Number of Segments** プロパティは無視されます。**Picture Properties**画面で **RESIZABILITY** プロパティをオンにすると、**Size X**、**Size Y** プロパティに合わせて拡大 / 縮小して表示するようになります。

18.5. つまみ類に関する **Connection** プロパティ




つまみ類のプロパティ画面には、MIDI コネクターをかたどったアイコンで示される、**Connection** ページがあります。

- ・ **RECEIVE MIDI-Activate MIDI In**: これをオンにすると、MIDI イベントを与えることにより、外部からつまみ類を調整できるようになります。イベントとしてはMIDI コントローラー・メッセージまたはポリ・アフタータッチ・メッセージを使え、それぞれコントローラー番号またはアフタータッチ・ノート番号を指定できます。
- ・ **Soft Takeover**: これをオンにすると、入力値が現在値より下から上に変化する、あるいは上から下に変化するまで、値が変化しないようになります。ハードウェアのつまみ類の動作に倣って、画面を切り替えたりスナップショットを呼び出したりした場合でも、値が突然大きく変化することのないようにする働きがあります。



Connection ページ。例はフェーダーの場合

- ・ **Incremental:** これをオンにすると、入力 MIDI メッセージは増分指定型のコントローラーから届いたものとして扱います。このようなコントローラーは、循環式ロータリー・スイッチ、CC(Continuous Controller; 連続式コントローラー) などともいい、Native Instruments の 4Control など多くの MIDI 機器に使われています。
- ・ **Panel to MIDI:** オンにすると、マウスでつまみ類を操作したときに MIDI イベントを送信するようになります。

- **Remote to MIDI:** オンにすると、MIDI イベントが与えられた結果値が変化したときに、MIDI イベントを送信ようになります (**Activate MIDI In** も参照)。シーケンサーが REAKTOR との間で MIDI データを送受信する場合、フィードバック・ループが生じる恐れがあるので注意してください。
- **Controller, Poly Aftertouch, MIDI Note:** MIDI メッセージによりつまみ類を制御する場合に、そのイベントの種類を指定します。
- **CONT/NOTE:** パネルの制御用に割り当てる MIDI コントローラー番号またはノート番号を指定します。
- **CONNECTIONS:** (フェーダー、ノブ、スイッチ、**XY** モジュール、ランプ、メーター、マルチ・ディスプレイ、マルチ・テキストの場合) MIDI 入出力系モジュールにも同じプロパティがあって、異なるインストールメント / マクロ間で、結線することなく信号経路を設定できます。さらに、OSC 接続を介して、別々のコンピューター間で信号をやり取りすることも可能です。内部的に接続するためには、制御元 (マスター) および制御先 (スレイブ) のつまみ類を下記の **OSC SOURCE**、**OSC TARGET** で指定し、それぞれ右上、右中のボタン (、) を押してください。すると、マスター側ならば「to」、スレイブ側ならば「from」という語に続ける形で、このリストに表示されます。1 つのマスターで複数のスレイブを制御することも可能で、この場合マスター側のリストには、つまみ類がいくつも列挙されることになります。なお、接続を解除したい場合は、右下のボタン () を押してください。
- **OSC SOURCE、OSC TARGET:** OSC 接続による制御元、制御先を選択するドロップ・ダウン・メニューです。**OSC SOURCE** には、OSC 経由でデータを送って来ている、制御元の花み類が並んでいます (ない場合は空のまま)。一方 **OSC TARGET** には、OSC でつながった他のコンピューター名が列挙されています。これは **OSC Setup** ウィンドウ (**System > OSC Settings** コマンドで開く) にあるリストと同じ内容です。
- **AUTOMATION-Disable:** オンにすると、プラグ・イン・ホストのパラメーター・オートメーション・リストに、選択肢としてこのつまみ類が表示されなくなります。

- ・ **ID: Disable** がオフの場合 (リストに表示する場合) に、その並び順を表す番号を指定します。なお、他のつまみ類と同じ番号を指定すると、番号が入れ替わります。但し、インストールメントの **Max Automation ID** プロパティーよりも大きな番号を指定すると、リストに表示されなくなるので注意してください。また、**XY** モジュール、マルチ・ディスプレイなど、オートメーション番号が2つ必要な場合は、ここで指定した番号に1を加えた値が第2の番号になります。したがってリストには、必ず続けて表示されることになります。

18.6. 制御パネルの編集

ストラクチャー・ウィンドウでモジュールを編集する場合と同様、制御パネル上のつまみ類も、**Duplicate** や **Delete** などのコマンドで編集できます。但し、つまみ類を複製あるいは削除した場合、ストラクチャーにも影響があるので注意してください。例えばあるつまみを削除すると、対応するモジュールも削除されてしまいます。したがって、こういった編集は、ストラクチャー側でのみ行うようお勧めします。

一方、位置を調整するだけならば、ストラクチャーには影響がありません。つまみ類のラベル上をつかんでドラッグしてください。配置がすべて決まったら、**Panel Lock** をオンにしておくといよいでしょう。これはコンテキスト・メニューにコマンドがあるほか、インストールメント・ヘッダーにある **Lock/Unlock Panel** ボタン (「スパナ」 ボタン)



でも切り替え可能です。ロックすると、ねじを締めた形のアイコンに変わります。

19. パネル操作

19.1. マウスによるつまみ類の操作

フェーダー



ハンドル部分をマウスでつかみ、上下 (あるいは左右) に動かしてください。

ノブ

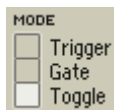


ノブをマウスでつかみ、上下に動かしてください。

ボタン

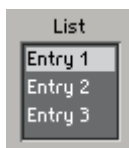


ボタンには、**MODE** プロパティの設定により、次の3種類の動作モードがあります。オン / オフの切り替え時に出力する値とともに、**Function** ページで設定してください。



- ・ **Trigger**: ボタンを押すと **On Value** 値のイベントが生成されます。ボタンを放す時点では何も生成されません。
 - ・ **Gate**: ボタンを押すと **On Value** 値、放すと **Off Value** 値のイベントが生成されます。
 - ・ **Toggle**: ボタンを押すごとに、2つの状態が交互に切り替わります。ボタンを押すと **On Value** 値のイベント、もう一度押すと今度は **Off Value** 値のイベントが生成されます。
- なお、オーディオ入力ポートにつないだ場合は、**Trigger** モードであっても **Gate** モードと同様に、ボタンを放した時点で **Off Value** 値になります。

リスト



該当する項目をクリックして選択します。

スイッチ



ボタンを押すと、それに対応する入力信号だけが通過、出力されるようになります。通過する信号は 1 系統だけで、2 系統以上の信号を同時に出力することはできません。

ドロップ・ダウン・メニュー (スイッチ、リスト)



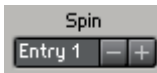
ドロップ・ダウン・メニューには、(そのままでは見えませんが、) いくつかの項目が並んでいます。この上でクリックすると、メニューが開いて各項目が表示されます (開いたらマウス・ボタンを放しても構いません)。その中から選択したい項目をクリックしてください。これはスイッチやリストの **STYLE** プロパティを **Menu** にした場合の動作です。

テキスト・パネル (スイッチ、リスト)



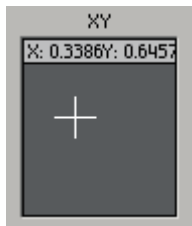
同様にいくつかの項目が並んでいますが、初めから各項目が表示されているので、1 回クリックするだけで選択できます。大きさを別に設定しますが、その大きさに全項目が入りきらない場合はスクロール・バーが現れます。これはスイッチやリストの **STYLE** プロパティを **Text Panel** にした場合の動作です。

スピン (スイッチ、リスト)



やはりいくつかの項目が並んでいます。右側の **+** ボタン、**-** ボタンを押して値を増減するほか、テキスト・パネルと同じように操作して選択することもできます。これはスイッチやリストの **STYLE** プロパティを **Spin** にした場合の動作です。

XY



2つのパラメーターを同時に制御できます。フィールド内でクリックし、上下に動かせばY、左右に動かせばXの値が変化します。

独自の制御パネル



スプリッター (REAKTOR に付属) の制御パネル

つまみ類の画像を別に用意し、「スキン」として設定することにより、機能はそのまま、外観を自由に変えることができます。さまざまなアンサンブルの設計者がそれぞれ工夫をこらしているの、どのようなしくみになっているのか、中身を調べてみるのも面白いでしょう。



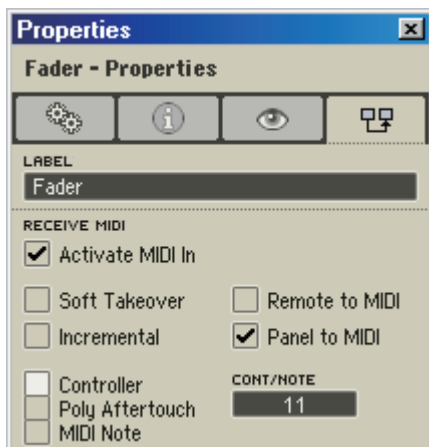
GoBox アンサンブル (REAKTOR に付属) の制御パネル

19.2. キーボードによるつまみ類の操作

フェーダー、ノブ、スイッチは、(コンピューター側の) キーボードでも操作できます。但し、あらかじめクリックして、選択状態にしておかなければなりません。

- ・ ↑ : 値を 1 単位増加、または前の選択肢に切り替え。
- ・ ↓ : 値を 1 単位減少、または次の選択肢に切り替え。
- ・ **page up**: 値を 10 単位増加。
- ・ **page down**: 値を 10 単位減少。

19.3. MIDI 信号によるつまみ類の制御



フェーダーのプロパティー画面、**Connection** ページ

- ・ **Activate MIDI In:** オンにすると、MIDI 信号でつまみ類を制御できるようになります。
- ・ **Controller:** MIDI コントローラー・メッセージで制御。
- ・ **Poly Aftertouch:** MIDI ポリ・アフタータッチ・メッセージで制御。
- ・ **MIDI Note:** MIDI ノート・メッセージで制御。ペロシティー値に応じてつまみ類の値が決まります。
- ・ **CONT/NOTE:** 制御に使う MIDI コントローラー番号または MIDI ノート番号。なお、コントローラー番号の場合は、下記の「MIDI Learn」機能を使って設定することもできます。

フェーダーやノブの場合は、MIDI メッセージの値によってその値が変わります。

ボタンを MIDI コントローラー / ポリ・アフタータッチで制御する場合は、値が 64 以上のときオンになります。また、MIDI ノート・オン / オフで切り替えることもできます。

スイッチやリストを MIDI コントローラー / ポリ・アフタータッチで制御する場合は、0 ～ 127 の範囲を入力ポート数で割った単位で値が変わります。例えば選択肢が 4 つある場合、0 ～ 31、32 ～ 63、64 ～ 95、96 ～ 127 のどの範囲の値かに応じて切り替わります。0 が一番下、127 が一番上の入力ポートに対応します。

MIDI Learn 機能

この機能のオン / オフは、ツールバー上の **MIDI Learn** ボタン



で切り替えます。外部 MIDI 機器から送られるメッセージと、これを使って制御するつまみ類の対応を設定するための機能です。

つまみ類を選択しておき、**MIDI Learn** ボタンを押してから、外部 MIDI 機器のホイール、ノブ、フェーダーその他を操作してください。これだけで対応づけの設定が終わります。

外部 MIDI 機器から送られてくるデータには標準型と増分型があって、通常は自動検出されますが、まれに誤認することがあります。その場合は **Incremental** プロパティを手動で修正してください。

- ・ **Incremental**: 増分型の MIDI コントローラー (循環式ロータリー・スイッチ) で値を制御する場合はオンにしてください。
- ・ **Soft Takeover**: フェーダーやノブを MIDI 信号で制御すると値が一瞬で大きく変化することがあり、振幅レベルを調整する場合など、雑音が生じるなどの悪影響が起こる恐れがあります。 **Soft Takeover** をオンにすると、入力値が現在値より下から上に変化する、あるいは上から下に変化するまで、値が変化しないようになるので、この影響を回避できます。

19.4. MIDI 出力

- ・ **Panel to MIDI**: オンにすると、制御パネル上でつまみ類を操作するのに応じて、所定の MIDI イベントが REAKTOR の MIDI 出力ポートから送信されるようになります。
- ・ **Remote to MIDI**: オンにすると、MIDI 経由で受信した MIDI イベントに応じて値が変わった場合にも、同じイベントが送信されるようになります。但し、シーケンサーを接続し、REAKTOR との間で MIDI データを送受信する場合は、フィードバック・ループが生じる恐れがあるので注意してください。

19.5. 独自のパネルの定義

独自の操作パネルを構築することも可能です。背景画像を用意しておき、操作に応じて表示を変化させることができます。矩形のビットマップに限りません。アルファ・チャンネルを利用して透明部分を定義すれば、どのような形にでも表示できます。つまみ類の設定は、**Snap**

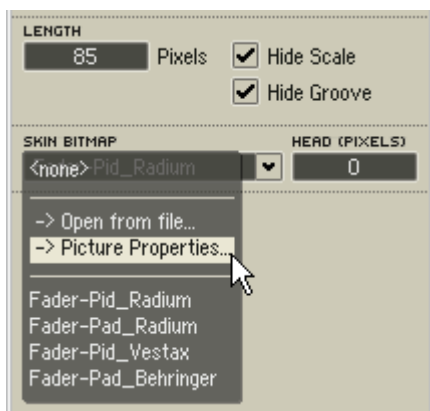
Value モジュールを使うと、スナップショットと組にして保存しておけます。

独自のフェーダー

インストゥルメント / アンサンブル用制御パネルの背景、インストゥルメントや基本マクロのアイコン、制御パネルのつまみ類、さらに、**Multi Display**、**Poly Display**、**Picture**、**Multi Picture** の各モジュールに使える画像形式としては、24 ビットの BMP 形式と、32 ビットの非圧縮 Targa(TGA) 形式があります。

Targa 形式の場合、一部を透明にする設定が可能です。

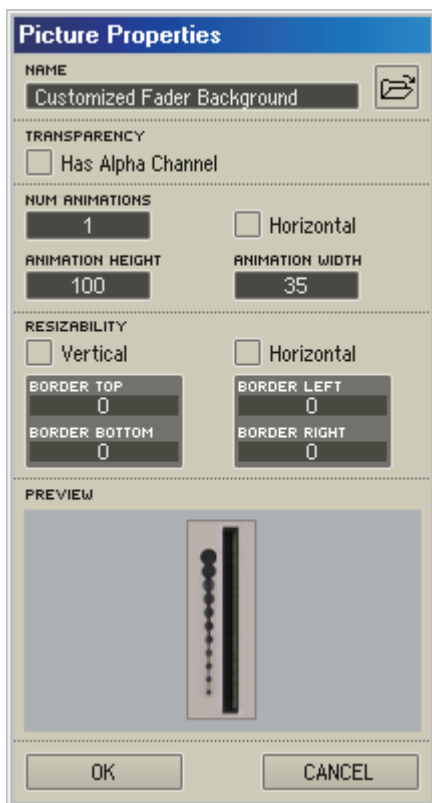
Picture Properties 画面



フェーダーのプロパティー画面、*Appearance* ページ

インストゥルメントやマクロ、**Picture**、**Multi Picture** などのモジュールは、プロパティー画面の **Appearance** ページに、**Select Picture/Select Object Picture/Select Background Picture/Select Structure Icon** というドロップ・ダウン・メニューがあります。選択肢の 1 つとして **Open from file...** という項目があり、これを実行すれば画像ファイルを読み込むことができます。そのほか、読み込み済みの画像ファイルがメニューの後半に並んでいます。

画像ファイルを読み込むと、**Picture Properties** 画面が開きます。画像に関する設定もここで行います。



Picture Properties 画面

注意：アンサンプルに読み込んだ画像は、これに属するどのオブジェクトでも使えます。同じ画像を再利用することにより、実質的な画像数をできるだけ減らして、消費メモリー量を節約してください。

読み込んだ画像ファイルに関する設定は **Picture Properties** 画面で行います。先に説明した **Select Picture** などのドロップ・ダウン・メニューから、**Picture Properties...** を選択してください。なお、ここで設定を変更すると、同じ画像を何箇所かで使っている場合、そのすべての表示に影響があります。

- ・ **NAME**: 内部的に画像を識別するために使う名前を表します。変更しても、ハード・ディスク中のファイル名が変わるわけではありません。

- ・ **TRANSPARENCY: Has Alpha Channel** をオンにすると、アルファ・チャンネルを使って画像の透明部分を定義することができるようになります。但し Targa(TGA) 形式の場合に限ります。アルファ・チャンネルによるマスクがかかっていない部分が透明になります。
- ・ **NUM ANIMATIONS:** フレーム数。2 以上の値を指定すれば、1 枚の画像をフレームに分割し、0 番から始まる番号で指定して表示することを表します。通常は縦方向に等分して使います。例えばノブの場合、最小値から最大値まで 128 段階の画像を縦に並べた画像を用意し、値に応じて表示を切り替える、という使い方をします。上から順に表示を切り替えていけば、最小値から最大値までノブが回っていく様子が表示されます。
- ・ **Horizontal:** オンにすれば、画像を縦方向ではなく横方向に分割して扱うようになります。但しあまり効率的ではないので、特に理由がなければオフにしておくようお勧めします。
- ・ **ANIMATION HEIGHT:** 縦分割モード (**Horizontal** がオフの状態) の場合、各フレームの高さ (ピクセル単位) を表します。
- ・ **ANIMATION WIDTH:** 横分割モード (**Horizontal** がオンの状態) の場合、各フレームの幅 (ピクセル単位) を表します。



ノブの *Picture Properties* 画面。画像を指定している様子

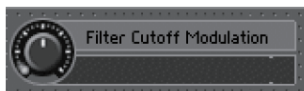
縦分割モードの場合、**NUM ANIMATIONS** と **ANIMATION HEIGHT** は、一方を指定すればもう一方も自動的に求められる、という関係になります。例えば画像の高さが 4000 ピクセルであれば、**NUM ANIMATIONS** を 40 と指定すると **ANIMATION HEIGHT** は自動的に 100 となります。逆に **ANIMATION HEIGHT** の方を 50 というように指定すると、**NUM ANIMATION** は 80 になります。

横分割モードの場合は、**NUM ANIMATIONS** と **ANIMATION WIDTH** が同様の関係になります。

- ・ **RESIZABILITY**: 大きな制御パネルの背景画像などの場合、小さな画像を縦横に並べて表示することにより、アンサンブル・ファイルの容

量を節約できます。また、つまみ類のスキンとして使う画像の場合は、オブジェクトの大きさに合わせて拡大 / 縮小する設定ができます。

- ・ **Vertical**: オンにすると、縦方向に画像を並べる、または拡大 / 縮小ようになります。
- ・ **Horizontal**: オンにすると、横方向に画像を並べる、または拡大 / 縮小ようになります。
- ・ **BORDER TOP、BORDER BOTTOM、BORDER LEFT、BORDER RIGHT**: 画像の一部を自分自身に重ね合わせるように表示する、という効果の設定ができます。X 軸または Y 軸に沿って同じ画像断片が一樣に延びている画像を、この方法で作ることができます。次の例は、**Horizontal** をオンにし、この機能を使って横に長いノブを表示したものです。

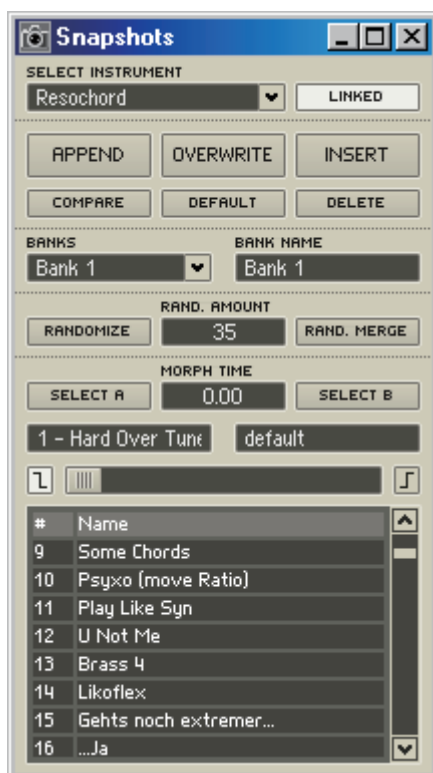


左: 元の画像

右: **RESIZEABILITY** の設定により拡大し、テキストを追加した様子

- ・ **PREVIEW**: 上記の設定を反映した状態で画像を表示します。

20. スナップショット



Snapshots ウィンドウ

「スナップショット」とはインストゥルメントの各種設定データをひとまとめにして保存し、いつでも呼び出して復元できるようにしたものです。ハードウェア・シンセサイザーでは「パッチ」、「プログラム」、「プリセット」などと呼ばれているものに相当します。制御パネル上のつまみ類の設定（ノブやフェーダーの値、リストやスイッチの設定、ボタンの状態など）や、MIDI コントローラーの設定を記憶しておきます。REAKTOR の場合、インストゥルメントごとに 2048 通りのスナップショットを保存でき、これを 16 のバンクに分けて管理しています。

注意：アンサンプルごとにはスナップショットを管理することも可能です。
詳しくは「スナップショットどうしのリンク」を参照してください。

ID FOR SNAPSHOT FILES

パネル上のつまみ類には一意的な番号がついており、スナップショットではこの番号でつまみ類を識別しています。具体的な値は、**Properties** ウィンドウの **Function** ページ、**ID FOR SNAPSHOT FILES** 欄で確認できます。



ID FOR SNAPSHOT FILES 欄


警告：この番号は変更しないでください。

この欄は編集できるようになっていますが、特に必要がない限り変更しないようお勧めします。というのも、スナップショットでは、この番号を基準にして値を管理しているからです。例えば「21 番のノブの値は 0.5、22 番のフェーダーの値は 0.75」という形です。この番号を交換してしまうと、次にスナップショットを呼び出したとき、21 番の値である 0.5 になるのはノブではなくフェーダーの方なので、折角保存しておいたものが役に立ちません。

別のインストールメント用に保存したスナップショットを呼び出した場合も、つまみ類と番号は対応していないので、結果がどうなるかはまったく予想できません。もっとも、これを逆用して思いがけない音を作る、という楽しみ方もあります。

スナップショットの呼び出し

スナップショットを呼び出して設定を復元する手順は 3 通りあります。

- ・ インストールメント制御パネル・ヘッダーの **Snapshots** ドロップ・ダウン・メニュー  から選択する方法。マウスで選択するほか、上下の矢印キーで切り替えることもできます。

- ・ **Snapshots** ウィンドウで選択する方法。このウィンドウは、**View > Show Snapshots** コマンドまたは **Snapshot** ボタン  で開きます。このとき、**SELECT INSTRUMENT** 欄で正しいインストゥルメントを選択することも忘れないでください。
- ・ MIDI プログラム・チェンジ・メッセージで呼び出す方法。但しあらかじめ **Recall by MIDI** プロパティをオンにしておく必要があります (**Function** ページ)。例えば値が 0 ならば 1 番、1 ならば 2 番のスナップショットが呼び出されます。

スナップショットの呼び出し、保存、ランダム化、補間 (モーフィング) を自動化する、**Snapshot** モジュールもあります。

スナップショットどうしのリンク

通常、スナップショットはインストゥルメント単位で管理します。例えば Inst1、Inst2 という 2 つのインストゥルメントから成るアンサンブルで、Inst1 のスナップショットを呼び出しても、Inst2 の方にはまったく影響がありません。

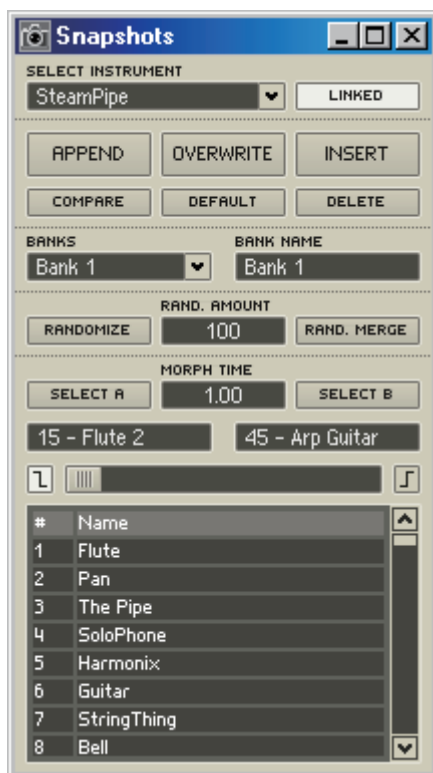
しかし、いくつかのインストゥルメントを組にして管理する方が便利な場合は、つぎの手順でスナップショットどうしをリンクしてください。

1. パネルセット・バーを使ってアンサンブル制御パネルを開き、この中にインストゥルメント制御パネルをすべて表示してください。
2. 各インストゥルメントについて、**Recall by Parent** プロパティをオンにしておきます (**Function** ページ)。
3. 既存のスナップショットを選ぶか、新たに作成してこれを選択状態にしてください。
4. インストゥルメントごとに、それぞれのスナップショットを選択してください。
5. この状態でアンサンブル・スナップショットを保存すると、次にこれを呼び出したとき、各インストゥルメントの状態がすべて復元されます。

スナップショットの管理

スナップショットの作成、保存、削除などは、専用の **Snapshots** ウィンドウで行います。**View > Show Snapshots** コマンドや、アンサン

ブル制御パネルのツールバーにある **Snapshot** ボタン  で開いてください。



Snapshots ウィンドウ

- ・ **SELECT INSTRUMENT**: 表示、編集したいインストゥルメントをここで選択します。なお、ここにはアンサンプルも選択肢として挙がっています。
- ・ **LINKED**: オンにすると、アンサンプル制御パネル・ウィンドウでインストゥルメントを選択するのに応じて、自動的に **Snapshots** ウィンドウの表示内容も切り替わるようになります。オフであればその都度 **SELECT INSTRUMENT** メニューで切り替えなければなりません。慣れてきたら、オンにした状態で作業するとよいでしょう。この下にある 6 つのボタンは、スナップショットの管理に使います。

- ・ **APPEND**: 現在の設定をスナップショットとして保存し、スナップショット・リストの末尾に追加します。スナップショット・バンクが満杯であれば空きのある別のバンクを探し、それでも見つからなければ新しいバンクを生成します。但し空きのスナップショット・スロットがない場合は何もしません。なお、インストールメント / アンサンブルごとに保存できるスナップショット数は最大 2048 で、これを 16 のバンクに分けて管理するようになっています。
- ・ **OVERWRITE**: 既存のスナップショットの内容を、現在の設定で上書きします。
- ・ **INSERT**: 現在の設定をスナップショットとして保存し、リスト中、選択状態のスナップショットの直下に挿入します。これより下にあったスナップショットは 1 つずつずれます。必要ならば次のバンクにずれていきます。

重要: **APPEND**、**OVERWRITE**、**INSERT** の各ボタンは、2 回押す必要があります。1 回押すとボタンが点灯し、追加 / 上書き / 挿入したスナップショット名の欄が、名前を入力できる状態になります。(必要ならば) 新しい名前を入力し、もう一度ボタンを押すと処理が完了します。2 回押さずにそのまま次の作業を始めてしまうと、思った通りの結果にならないので注意してください。

- ・ **COMPARE**: 現在の設定と、スナップショットの設定とを比較できます (詳しくは後述の「スナップショットの比較」を参照)。
- ・ **DEFAULT**: スナップショットの内容を、あらかじめ **Properties** ウィンドウで設定しておいたデフォルト状態に戻します。
- ・ **DELETE**: スナップショットを削除します。リストには空のスロットがあるので、必要に応じて **BANKS** メニューの **Sort** コマンドで並べ替えてください。

スナップショットの名称変更、コピー

スナップショット名を変更する手順は次の 2 通りがあります。

- ・ スナップショット名の欄をダブル・クリックし、新しい名前を入力して **return** キーを押す方法。
- ・ スナップショットを選択して **OVERWRITE** ボタンを押し、新しい名前を入力した後もう一度 **OVERWRITE** ボタンを押す方法。
コピーする手順も 2 通りあります。

- ・ スナップショットを選択して **APPEND** ボタンを押し、新しい名前を入力した後もう一度 **APPEND** ボタンを押す方法。リスト上、最初に見つかる空きスロットに追加されます。
- ・ スナップショットを選択して **INSERT** ボタンを押し、新しい名前を入力した後もう一度 **INSERT** ボタンを押す方法。コピー元スナップショットの直下に追加されます。

スナップショットの比較

COMPERE ボタンには次の 2 つの役割があります。

- ・ 同じスナップショットの修正前と修正後を比較すること。設定を少しずつ改善していく作業に役立ちます。
- ・ 別々のスナップショットを比較すること。

しくみは単純で、一方のスナップショットを比較専用のバッファーに保存し、**COMPERE** ボタンを押すごとに、現在の設定とこのバッファーの内容を入れ替えるようになっているだけです。

同じスナップショットの修正前と修正後を比較する手順：

1. **Snapshots** ウィンドウで、スナップショットを選択してください。
2. **COMPARE** ボタンが点灯していないことを確かめておきます。
3. つまみ類の設定を修正してください。
4. ここで **COMPARE** ボタンを 2 回押すと、一度点灯し、再び消灯します。これにより、修正後の状態が比較バッファーに保存されました。
5. これ以降、**COMPARE** ボタンを押すと、修正前と修正後の状態がその都度入れ替わるので、どこが違っているか見比べることができます。
6. さらに設定を修正する場合は、2～5 を繰り返してください。

別々のスナップショットを比較する手順：

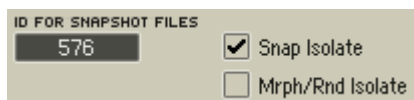
1. **Snapshots** ウィンドウで、スナップショットを選択してください。
2. 比較するもう一方のスナップショットを選択すると、先に選択しておいた方が比較バッファーに保存されます。
3. これ以降、**COMPARE** ボタンを押すと、2 つのスナップショットの状態がその都度入れ替わるので、どこが違っているか見比べることができます。

スナップショットの修正中に誤って別のスナップショットを選択してしまっても、さらに修正を施す前であれば、**COMPARE** ボタンを押して修正前の状態に戻すことができます。

Snap Isolate

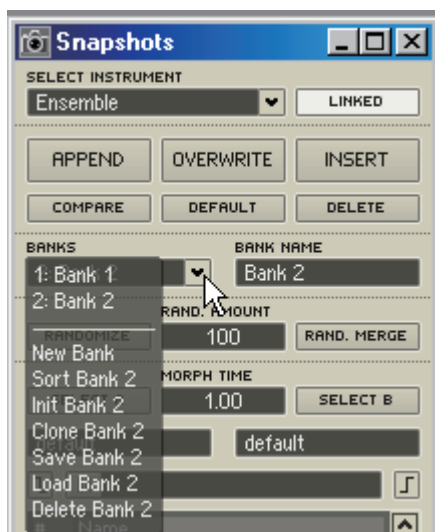
スナップショットを切り替えるとつまみ類の設定値は変わってしまうわけですが、そうしたくないこともあります。例えばシーケンサーの制御パネルで、テンポ調整用のノブだけは値が変わらないようにしたい、というような場合です。

そのような場合は、該当するつまみ類の **Snap Isolate** プロパティー (Function ページ) をオンにしてください。



Snap Isolate の設定。Properties ウィンドウの Function ページ

BANKS メニュー



Snapshots ウィンドウの BANKS メニュー

BANKS メニューは2つの部分に分かれています。上側にはバンク名、下側にはバンクの管理コマンドが並んでいます。

- ・メニューの上側でバンクを選択すると、**Snapshot** メニューにはそのバンクに属するスナップショットが列挙されます。
- ・バンク名を変更したい場合は、メニューの右側にある **BANK NAME** 欄をダブルクリックし、新しい名前を入力してください。
下側には次のようなコマンドが並んでいます。
- ・ **New Bank**: バンク・リストの最初の空きスロットに、新しいバンクを作ります。例えば Bank 1 と Bank 3 があるインストゥルメントで **New Bank** を実行すると、Bank 2 ができます。Bank 1、Bank 2、Bank 3 があった場合は Bank 4 ができることになります。インストゥルメント / アンサンブルごとに、最大 16 のバンクが作れます。
- ・ **Sort**: スナップショットを番号順に並べ替え、空きスロットを詰めます。
- ・ **Init**: バンクに属するスナップショットをすべて削除します。
- ・ **Clone**: バンクの複製を作り、最初の空きスロットに保存します。
- ・ **Save**: バンクに属するスナップショットを、*.ssf 形式のファイルとして保存します。
- ・ **Load**: バンクに属するスナップショットを、*.ssf ファイルから読み込みます。このとき、既存のバンクに上書きする形で読み込むか、新しいバンクとして追加するかを選択ができます。
- ・ **Delete**: バンクを削除します。

誤ってバンクを削除してしまっても、**Undo** コマンドで取り消せます。

スナップショットのランダム化

つまみ類の設定値を、所定の幅でランダム化する機能です。



Snapshots ウィンドウのランダム化関連ボタン

- ・ **RANDOMIZE**: スナップショット上のつまみ類の設定がすべてランダム化されます。但し後述の **Random Isolate** がオンになっているものを除きます。
- ・ **RAND. AMOUNT**: ランダム化の程度を 0 ～ 100% の範囲で設定します。すなわち、現在の設定値の上下、この比率で決まる範囲のランダムな値になります。例えば、値の範囲が -1 ～ 1 のノブを考えま

しょう (最大値と最小値の差は 2)。現在値が 0 であるとし、**RAND. AMOUNT** を 25% としてランダム化を施すと、-0.5 ~ 0.5 の間のあるランダムな値になります ($0 \pm (2 \times 25\%)$)。一方、現在値が -0.5、**RAND. AMOUNT** が 50% であれば、-1 ~ 0.5 の間になります ($-0.5 \pm (2 \times 50\%)$)。このように、最小値~最大値の範囲を外れることはありません。

- ・ **RAND. MERGE**: 後述のモーフィング機能と組み合わせて使います。それぞれのつまみ類について、2 つのスナップショットにおける設定値の間の値を割り当てて徐々に変化させるのですが、その変化させる割合をランダムに決める機能です。ランダム化の程度は **RAND. AMOUNT** で決まります。例えば 50% であれば、25 ~ 75% の範囲、すなわち 2 つのスナップショットの中間辺りの状態になります。一方、100% とすると、スナップショット間のどの状態にもなりえます。

ランダム化の対象としたくないつまみ類は、**Random Isolate** プロパティをオンにしておいてください (**Function** ページ)。

スナップショット間のモーフィング機能

あるスナップショットから別のスナップショットの状態まで、それぞれのつまみ類の設定値を徐々に変化させていくことができます。





Snapshots ウィンドウのモーフィング関連ボタン

モーフィングの手順:

1. スナップショット A から B までの変化にかかる時間を、**MORPH TIME** 欄に、0 ~ 60 秒の範囲で指定します。
2. **SELECT A** ボタンを押し、一方のスナップショットをリストから選択してください。
3. 同様に、**SELECT B** を押しもう一方のスナップショットを選択してください。

4. スライダーを動かすと、制御パネルの状態が変わります。一番左の位置ならばスナップショット A と同じ状態ですが、中央位置にすると 2 つのスナップショットのちょうど中間の状態になります。

注意：スライダーを動かしても一瞬で状態が変わるのではなく、**MORPH TIME** で指定した時間をかけて徐々に変化していきます。

ボタンやスイッチの場合は値が徐々に変化するということがないので、モーフィングの対象にはなりません。あらかじめ、どちらのスナップショットの状態にするか、スライダーの左右にあるボタン (、) で指定しておいてください。採用した方のボタンが点灯します。

21. サンプリングと再合成

この章では、REAKTOR に組み込まれた強力な機能である、オーディオ・サンプルを材料とした音声再合成について解説します。

21.1. サンプルの管理

サンプル・ファイルと RAM

REAKTOR のサンプラー系モジュールは、WAV 形式または AIFF 形式のオーディオ素材を扱えます。モノラル / ステレオのどちらにも対応しており、サンプル・レートやビット・レートの制約もありません。また、ループやキーボード割り当てに関するデータも認識できます。

音声ファイル进行处理するためには、まず RAM 上に読み込んでおく必要があります。処理効率を考慮して、音声ファイル自身のビット・レートとは関係なく、常に 32 ビット形式に変換して処理するようになっています。CD と同等のステレオ音声の場合、1 分あたり 20MB の RAM を消費します。仮想メモリーを使えば物理メモリー容量よりも多くのデータを読み込みます (Windows ではこれが有効になっていますが、MacOS ではそうではありません)。仮想メモリーの管理は OS 側で行われ、REAKTOR 側で認識することはできないので、物理メモリー容量を超える巨大な音声ファイルを読み込んでも、その時点ではエラーになりません。

しかし後になってから、CPU が過負荷のため処理できない、というエラーが生じる場合があります。データが RAM 上にあるように仮想メモリーのしくみを使って見せかけているけれども、実際には (低速の) ハード・ディスク上にあるので、サンプル・データに対する処理が間に合わなくなってしまうのです。エラー・メッセージが出ないとしても、音声データ・ストリームが中断されるため、ぶつぶつと音が途切れる現象となって現れます。特に生演奏中にこのようなことが起こると具合が悪いので、メモリーを十分に搭載してください。

同じサンプルの重複使用

あるサンプラー系モジュールに読み込んだのと同じサンプルを別のモジュールでも読み込んだ場合、実際には改めて RAM 上に読み込むこと

なく、同じものを共用します。したがって RAM の消費量が増えることはありません。同様に、モジュールの声道数を増やしても、それに比例して RAM の消費量が増えるとは限りません。

同じサンプルであるかどうかは、音声ファイルのパス (ディレクトリーとファイル名) を比較して判断します。すなわち、音声ファイルを読み込む際、同じパスのファイルを既に読み込んでいれば、改めて読み込むことなく再利用するようになっています。

サンプルの欠落

サンプルを扱うインストゥルメントを保存する場合、音声データそのものを埋め込むことなく、対応する音声ファイルのパス名だけを保存するようになっています。したがって、音声ファイルを削除、改名、移動すると、再生できなくなってしまう。

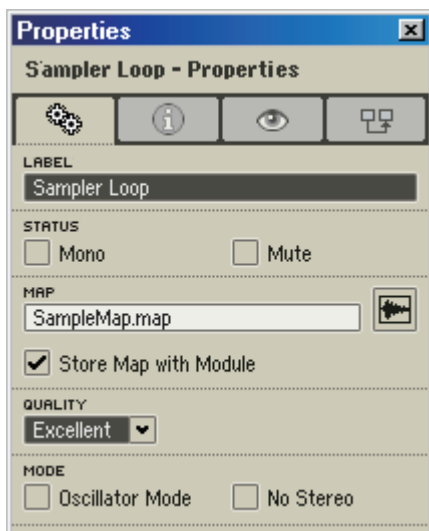
このような場合、インストゥルメントを開こうとしたときにエラー・メッセージが現れます。サンプル・マップ・エディターでリスト表示すると、欠落しているサンプルにはその旨の印がつきます。**Replace** ボタンを押してファイル選択ダイアログ・ボックスを開き、音声ファイルを選択し直してください。

モジュール中にサンプルを埋め込んで保存

サンプラー系モジュールで **Store Map with Module** プロパティー (**Function** ページ) をオンにすると、サンプル・データそのものを埋め込んだ形でアンサンプル・ファイルに保存するようになります。次に読み込む際は、元のファイルの有無に関係なく、いっしょに保存しておいたデータを参照します。

そのため、元のファイルを移動 / 削除したとしても問題が起こりませんし、ほかのコンピューターに転送する際に洩れが生じることもなくなります。但し、アンサンプル・ファイルの容量が非常に大きくなってしまいます。

繰り返しになりますが、**Store Map with Module** がオフのままに保存したアンサンプルを共用する場合、サンプル・ファイルを同じ場所に置いておかねばなりません。ライブラリーに登録して他の人とも共用するアンサンプルは、忘れずにオンにしておいてください。



サンプラー系モジュールのプロパティー画面、**Function** ページ

サンプルの解析

Grain Resynth、**Grain Pitch Former**、**Beat Loop** などのサンプラー系モジュールは、サンプル・ファイルをもとに、リアル・タイムで音声を再合成します。そのため、サンプルの読み込み時に中身を分析し、必要なデータを求めておくようになっています。この処理にはサンプルの長さ按比例した時間がかかります。読み込みのたびに分析するのは無駄なので、得られたデータをサンプル・ファイルに埋め込む形で保存しておくことも可能です。元のファイルを書き換えてしまうことになるので、事前にその旨を表示し、保存するかどうか訊ねるようになっています。もちろんファイルが書き込み保護されている場合は保存できません。

注意：いったん分析データを書き込んでしまうと、元になったサンプルを書き換えても同じ分析データを使おうとする、という問題があります。その場合は、ファイル名を変更してからモジュールに読み込んで、強制的に分析し直してください。

サンプル・エディター

REAKTOR にはサンプルの編集機能が組み込まれていないので、他のアプリケーションを使って編集することになります (Windows ならば Sound Forge、WaveLab、Audition、GoldWave など、MacOS X ならば Peak、Spark XL、Audacity など)。編集作業の便宜のため、サンプル・マップ・エディター上に開いたサンプルを、所定のエディターで編集できるようになっています。

1. サンプルの編集に使うエディターの場所を、環境設定画面の **Directories** ページ、**External Sample Editor** に指定しておいてください。
2. サンプル・マップ・エディター上で、サンプル・ファイルを選択してください。
3. **Edit Sample List > Edit** コマンドを実行すると、サンプル・エディターが起動されます。
4. 必要な編集を施し、保存してください。
5. サンプル・マップ・エディターに戻り、サンプルを選択して **Edit Sample List > Reload** コマンドを実行すると、編集後の新しいサンプルに置き換わります。

注意：サンプル・エディターによっては、機能の制限のため、保存するとループ情報がなくなってしまう場合があります。

21.2. サンプル・マップ

(電子音ではない) アコースティック楽器の音をシミュレートする、というのがサンプル・マップの用途としては最も多いでしょう。この場合、音域に応じていくつかのサンプルを割り当てて使うことになります。例えばクラリネットの音域は 3 オクターブに及びますが、高音域と低音域では音色がかなり異なるので、十数個程度の音域に分けてサンプルを用意して使い分けるのです。1 つのサンプルを機械的に移調して広い音域で使うのでは、楽器の音を自然に再現できません。

さらに、同じキーに対しても、強さに応じてサンプルを使い分けることが可能です。以下、それぞれの使い方について詳しく説明します。

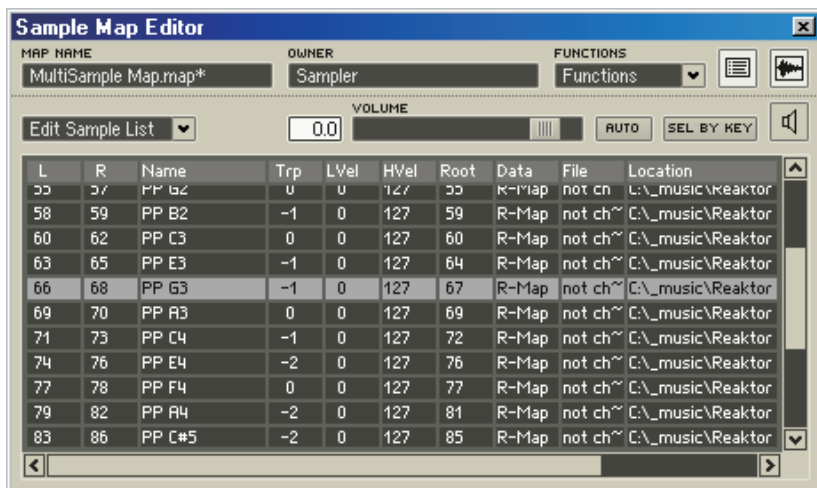
マルチ・サンプリング

何種類もの音の生成に同じサンプルを繰り返し使うと、本来の楽器らしさが損なわれてしまいます。そのため REAKTOR には、複数のサンプルを用意しておき、切り替えながら再生する機能があります。

通常、あるサンプルを元の音高とは異なるキーに割り当てた場合、本来の高さから離れるほど不自然な響きになっていきます。これは、単純に高さ（周波数）を変えたとしても、倍音特性は変わらないからです。そこで、音高に応じていくつものサンプルを用意し、音域によって使い分けることによりこの問題を解決しています。

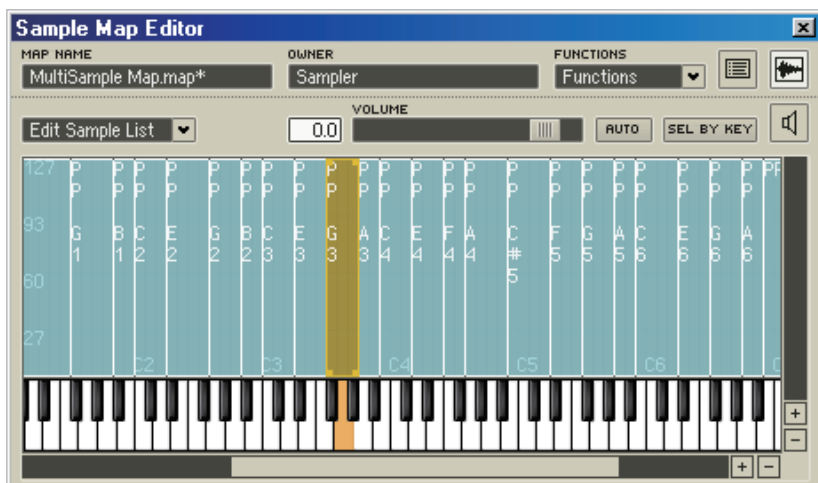
複数のサンプルを用意してサンプル・マップを作成する場合、各サンプルについて、少なくとも次の3つのパラメーターを設定しなければなりません。設定にはサンプル・マップ・エディターを使います。

- ・ **Root**: サンプルそのものの音高です。
- ・ **L**: そのサンプルを使う音域の最低音を表します。
- ・ **R**: そのサンプルを使う音域の最高音を表します。



サンプル・マップ・エディターにいくつものサンプルが並んでいる様子

Root が **L** と **R** のちょうど中央辺りになるようにし、1つのサンプルでカバーする音域を広くし過ぎないようにするとよい結果が得られます。



サンプル・マップ・エディターの一部。橙色の帯が **Root** に相当

サンプラー系モジュールには必ず、ピッチを指定する **P** 入力があります。通常、ここには **Note Pitch** モジュールをつないで、MIDI ノート番号を与えます。この値を基準に、サンプル・マップから適当なサンプルを選択し、ピッチを変換して出力します。

ピッチとは独立にサンプルの切り替えを制御したい場合は、やはりサンプラー系モジュールには必ずついている、**Sel** 入力を使います。アコースティック楽器とはかなり異なる響きになりますが、それはそれで面白い効果が得られる場合もあります。

さらに、1つのキーに複数のサンプルを対応させることも可能です。これは例えば、強弱(デュナーミク)による響きの違いを再現するために有用です。MIDI の場合はベロシティ値に応じてサンプルを使い分けることになります。そのためには、あらかじめ強さの異なるサンプルをいくつか用意しておきます。演奏時には、ベロシティ値に応じてサンプルを使い分けることにより、元の楽器の響きをより忠実に再現できるのです。

ドラム・マップ

「ドラム・マップ」の場合もマルチ・サンプリングと同様に、**Root**、**L**、**R** のパラメーターで、あるサンプルを割り当てる音域を設定します。もっ

とも、元々「高さ」が明確な音ではなく、音高を変換することもないため、**Root** 自身には意味がないのが普通です。

P 入力に与えるピッチとは独立に、**Sel** 入力でサンプルの切り替えを制御できる点も同様です。例えば **Gate** モジュールを接続すると、ベロシティーに応じてサンプルが切り替わるので、MIDI キーボードを弾く強さに応じて音の種類を変えることができます。

マップの保存

サンプル・マップは、インストゥルメントやアンサンブルとは別に保存しておくことができます。Windows の場合、ファイル名には **.map** という拡張子がつきます。


サンプル・データ自身をファイルに格納する方法と、サンプル・ファイル名だけを記録しておく方法があり、**Store Sample with Ensemble** プロパティーで切り替えます。

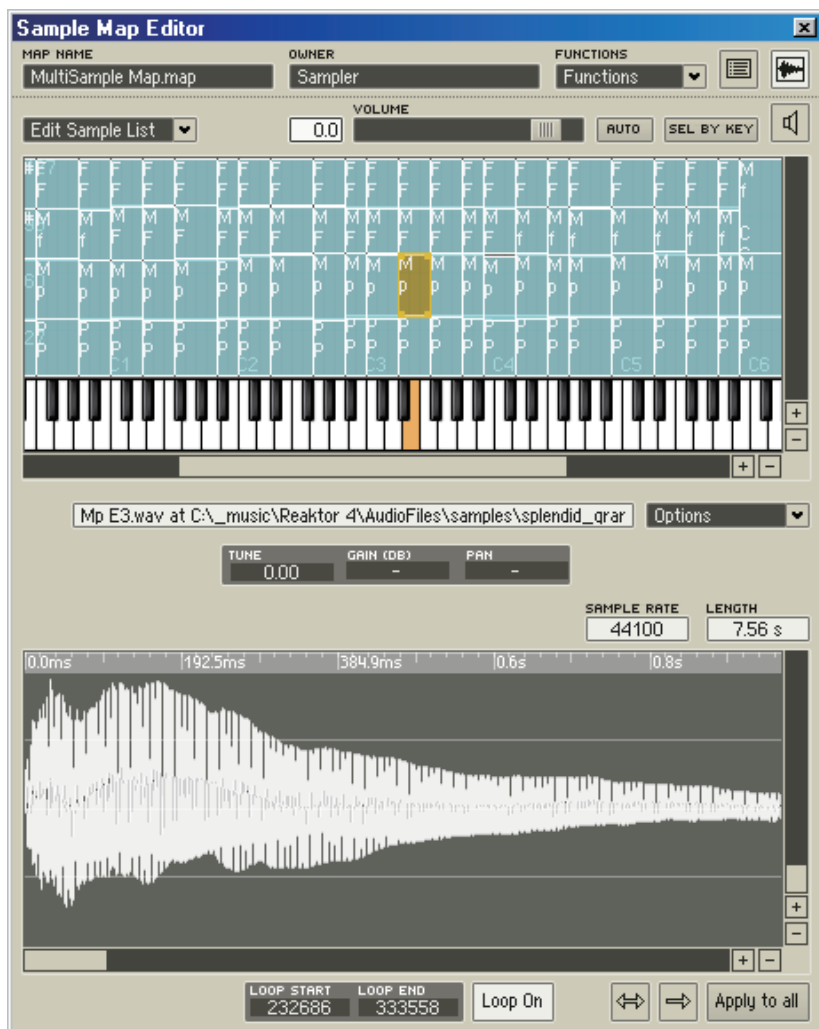
21.3. サンプル・マップ・エディター

サンプル・マップ・エディターは、**Sampler**、**Grain Resynth**、**Beat Loop**、**Sample Lookup** などのサンプラー系モジュールで使う音声ファイルの読み込み、保存、編集、音域とサンプルの対応づけ (マッピング)、ループ設定などを行います。エディター画面を開いておくと、サンプラー系モジュールを選択することに画面表示も切り替わります。サンプラー系モジュールを何も選択していない場合は何も表示されません。

サンプル・マップ・エディターを開く手順

サンプル・マップ・エディターは、主メニューの **View > Show Map Editor** コマンドで起動します。この時点では、サンプル・ファイルを何も開いていない状態です。

サンプラー系モジュールのプロパティー画面、**Function** ページにある **Show Map Editor** ボタン  を押すと、このモジュールに割り当てられたサンプル・ファイルがすべて開いた状態で立ち上がります。モジュールに対応する制御パネル上でコンテキスト・メニューを開き、**Open map Editor** コマンドを実行しても同様です。

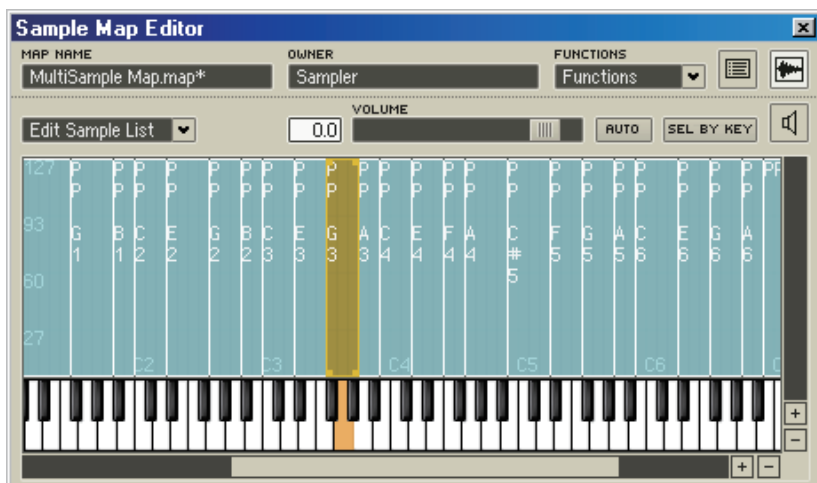


複数のサンプルから成るサンプル・マップを表示している様子

エディターの画面構成

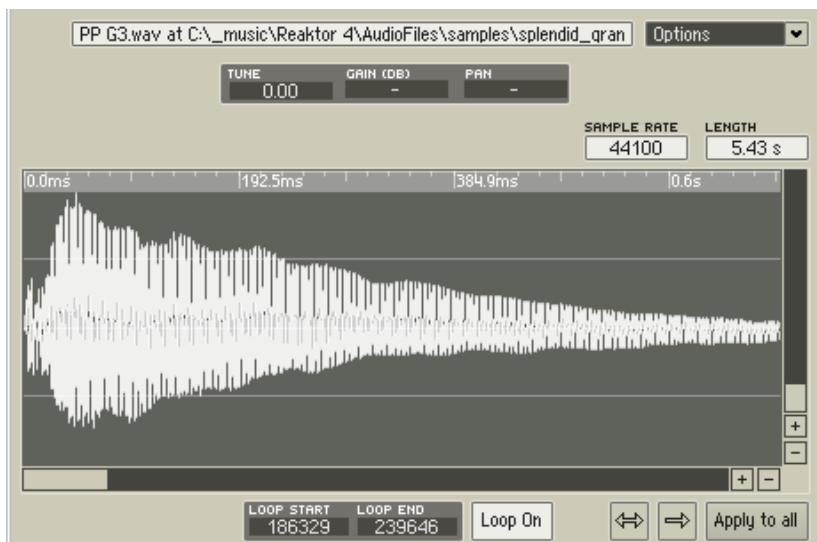
サンプル・マップ・エディターは2つの部分に分かれています。

- ・「サンプル・マップ」部分(上側): 個々のサンプルを読み込んでマップ上に配置するための領域です。



サンプル・マッパーの部分 (上側)

- ・「ループ・エディター」部分 (下側): ループ点の編集に使います。




ループ・エディターの部分 (下側)


それぞれについて詳しく見てみましょう。

サンプル・マッパー

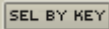
サンプル・マッパーには次のようなボタン類があります。

- **MAP NAME:** サンプル・マップ・ファイル名。編集も可能です。
- **OWNER:** このマップを使うサンプラー系モジュールの名前。
- **FUNCTIONS** メニュー：**Remap to Single Keys**、**Set transpose to Null for All** のコマンドを起動できます (詳しくは後述)。

-  **Map View** ボタン: サンプル・マッパーにはリスト表示と図形表示 (キーボード上にマップを表示) がありますが、その切り替えに使います。

-  **Show/Hide Loop Editor** ボタン: ループ・エディターの表示 / 非表示を切り替えます。

- **Edit Sample List** メニュー: サンプルの編集コマンドを実行します。

-  **SEL BY KEY** ボタン: オンにすると、MIDI ノート・メッセージを与えたとき、これに応じてサンプルの選択状態が切り替わるようになります。

-  サンプル試

聴用のボタン類: 詳しくは「サンプル・ファイルの試聴」を参照。

- (サンプル名): サンプル・ファイル名とその場所 (ディレクトリー) を表示します。

- **Options** メニュー: **Auto-Move RootKey**、**Ignore RootKey When Loading From File**、**Single Key Mode** の各コマンドがあります。詳しくは「Options メニュー」を参照。

-  **TUNE、GAIN (DB)、PAN:**

サンプルのピッチ調整幅 (セント単位)、利得 (dB 単位)、定位 (ステレオ場上の位置) を表示します。

-  **SAMPLE RATE、LENGTH:** サンプル・レートと長さ (ミリ秒単位) を表示します。

Edit Sample List メニュー

ここには次のようなコマンドが並んでいます。

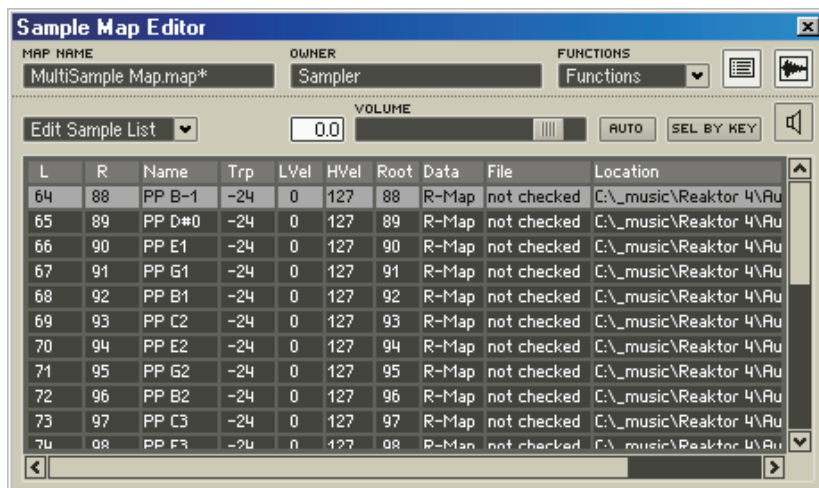
- ・ **Add:** あるサンプルをサンプル・マップに追加します。
- ・ **Replace:** あるサンプルを別のサンプルで置き換えます。
- ・ **Save:** サンプルをファイルに保存します。
- ・ **Delete:** あるサンプルをサンプル・マップから削除します。サンプル自体がハード・ディスクから削除されるわけではありません。
- ・ **Edit:** 他のオーディオ・エディターを起動してサンプルを編集できるようにします。なお、起動するエディターは、環境設定画面の **Directories** ページで設定しておいたものです。
- ・ **Update:** サンプルをファイルから読み込み直します。他のエディターで編集した後で実行してください。

サンプルを何も選択していない場合、**Delete**、**Edit**、**Update** の各コマンドは、マップ全体を対象として働きます。

- ・ **Load Map:** サンプル・マップ・ファイルを読み込みます。
- ・ **Save Map:** サンプル・マップをファイルに保存し、他のサンプラー・モジュールでも使えるようにします。このとき、サンプル・データもマップに埋め込む形で保存するかどうか訊ねられます。**Yes** と応えた場合、ファイルの容量は大きくなりますが、そのままほかの環境に持っていても問題は起こりません。一方、**No** と応えた場合は各サンプルのファイル名とディレクトリーを保存するだけなので、マップ・ファイルだけをほかの環境に持ってってしまう、といったことのないよう注意しなければなりません。
- ・ **Akai Import:** **Akai Import** ウィンドウを開きます。これは AKAI S1000/S3000 形式の CD からサンプル・マップをインポートするために使います (詳しくは後述)。

Tab キー、**Shift+Tab** キーで、サンプルを順次切り替えることができます。

サンプル・マップのリスト形式表示



リスト形式で表示したサンプル・マップ

リスト形式に切り替えると、サンプル・マップに属する各サンプルの情報が行ごとに並びます。見出し欄をクリックするとその列を基準に昇順で並び、もう一度クリックすると降順になります。

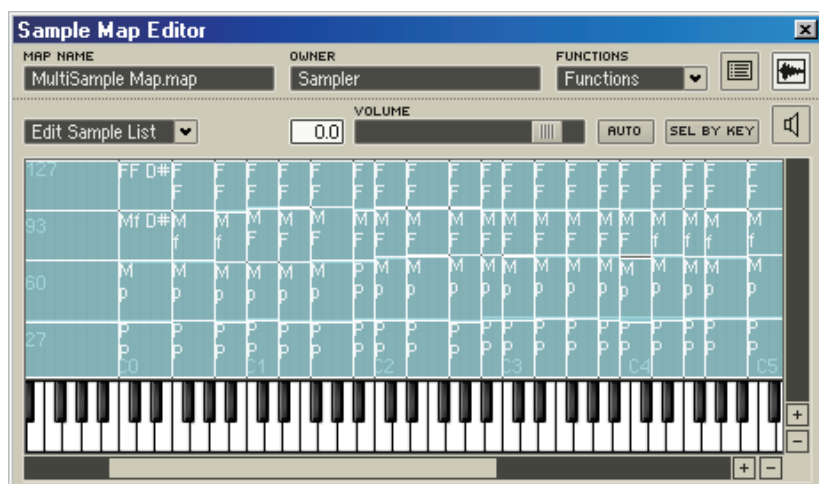
次の各データは、「編集不可」と書いてあるものを除き、編集可能です。

- ・ **L**: このサンプルでカバーする音域の最低音 (MIDI ノート番号)。
 - ・ **R**: このサンプルでカバーする音域の最高音 (MIDI ノート番号)。
- L** と **R** で、このサンプルを使う音域を表します。これはグラフィック形式の方が分かりやすいでしょう。
- ・ **Name**: サンプル名。編集不可。
 - ・ **Trp**: サンプルの移調幅。 **Root** の音高を半音いくつ分上下にずらせば **L** で表される音高になるか、その差を表します。この値を編集すると、 **L** の値は変わらないままで、 **Root** の値が変わります。
 - ・ **LVel**: このサンプルでカバーするベロシティー範囲の最小値。
 - ・ **HVel**: このサンプルでカバーするベロシティー範囲の最大値。
- LVel** と **HVel** で、このサンプルを使うベロシティー範囲を表します。同じ高さの音でも、強さによって使うサンプルを区別したい場合に指定します。
- ・ **Root**: サンプルをそのまま再生したときの音高 (MIDI ノート番号)。

- ・ **Data:** 編集不可。
- ・ **File:** 編集不可。
- ・ **Location:** サンプル・ファイルの、ハード・ディスク上の位置。編集不可。

注意：音域やベロシティー範囲が重複するサンプルがある場合、どちらが優先されるかはグラフィック形式表示で確認できます。一般に、後から追加した方が優先になります。

サンプル・マップのグラフィック形式表示



グラフィック形式で表示したサンプル・マップ

グラフィック表示にすると、画面の上半分には各サンプルがカバーする範囲を表す 2 次元の領域が表示されます。横軸は音域を表し、ピアノ鍵盤の形で表示します。MIDI ノート・イベントの、ノート番号 (横軸) およびベロシティー値 (縦軸) の範囲に応じ、どのサンプルが使われるかを表しています。この区画上にマウス・カーソルを持っていき、四方向の矢印型カーソルになった状態でドラッグすると区画を移動できます。また、区画の境界付近にマウス・カーソルを持っていき、二方向の矢印型カーソルになった状態でドラッグすると、範囲を伸縮することができます。

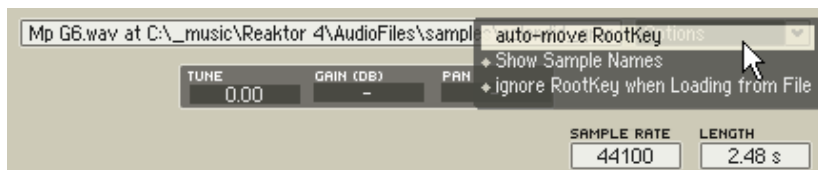
複数の区画をまとめて選択、編集することも可能です。移動や伸縮の操作をすると、どの区画に対しても一斉に適用されます。

区画が重なり合っても構いませんが、これは編集で一時的にそうするだけにとどめてください。同時に複数のサンプルを再生することはできないので、最終的には重複をなくする必要があります。

Functions メニュー

- ・ **Remap to Single Keys:** マップ上の各サンプルをすべて (選択しているか否かにかかわらず)、カバーする音域の最低音 1 つだけに割り当て直すために使います。マップ上のサンプルの並び順はそのまま保存されます。
- ・ **Set Transpose To Null For All:** 各サンプルの移調幅を一斉に 0 にリセットするコマンドです。これに応じて **Root** の値も変わります (音域の最低音から移調幅を引いた値)。これにより、区画の最低音がもっとも自然なピッチで響くようになります。この効果はリスト表示に切り替えると分かりやすいでしょう。

Options メニュー



サンプル・マップ表示の下には、選択状態のサンプルの名前と、ハード・ディスク上の位置が表示されています。その右側にある **Options** には、次のような選択肢が並んでいます。

- ・ **Show Sample Names:** オンにすると、サンプル・マップ上にサンプル名が表示されます。
- ・ **Ignore Root Key when Loading:** オンにすると、サンプル・ファイルにルート・キー情報が記録されていてもそれを無視するようになります。
- ・ **Single Key Mode:** オンにすると、サンプルがカバーする音域の最低音を変更したとき、最高音も自動的に同じ値になります (リスト表示の場合のみ有用)。これは編集の手間を減らすための機能です。

サンプル・ファイルの試聴



読み込んだサンプルは、次の手順で試聴できます。

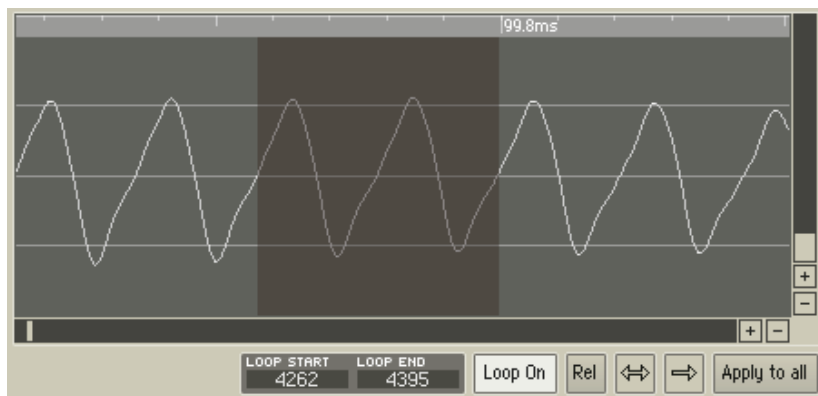
1. 試聴したいサンプルを選択してください。
2. **Play** ボタン（スピーカーをかたどったアイコン）で再生 / 停止を切り替え、**VOLUME** フェーダーで音量を調整します。
3. **AUTO** ボタンをオンにすると、サンプルを選択したとき、自動的に再生が始まるようになります。

ループ・エディター

画面下半分のループ・エディターは、各サンプルのループ点を設定、編集するために使います。表示されていない場合は、画面右上の **Show/Hide Loop Editor** ボタン XXXXXX を押してください。波形表示のうち濃い茶色の部分がループ範囲を表します。

ループの終了点は、ループ範囲の右端をマウスでつかんでドラッグすれば変更できます。また、中央付近をつかんでドラッグすれば、範囲全体が前後に動きます。なお、ループ範囲の全体を表示するためには、右下の **+/-** ボタンで拡大 / 縮小する必要があるかも知れません。

開始点と終了点は波形表示の下 (**LOOP START**、**LOOP END** に数値でも表示されますが、ここで直接編集することはできません。



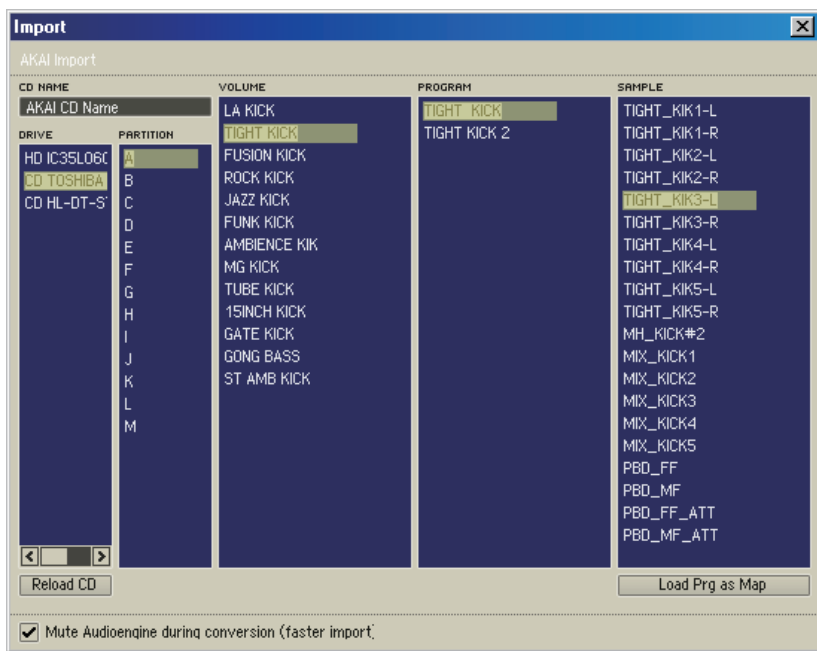
ループ・エディター (サンプル・マップ・エディターの下側部分)

その下には、ループ点に関するプロパティを編集する、5つのボタンが並んでいます。

- ・ **Loop On**: ループのオン / オフを切り替えます。
- ・ **Rel**: 「ループ・イン・リリース」モードにします。これは、ゲート信号がオフになってもループを継続する、というモードです。
- ・ ⇄: ループの反転。末尾まで再生すると、今度は先頭に向かって逆方向に進むようになります。つまり、方向を反転しながら交互に再生されるわけです。ループの継ぎ目が目立たないようにする効果があります。
- ・ ⇒: 反転再生。サンプルを常に逆方向に再生します。
- ・ **Apply to all**: ループの設定をマップ上のサンプルすべてに一斉に適用します。

21.4. Akai ファイルのインポート

AKAI S1000/S3000 形式のファイルを、サンプル・マップ・エディター上から、あるいはサンプラー系モジュールの制御パネルのコンテキスト・メニューを使ってインポートすることができます。Akai CD をドライブに挿入しておき、**Akai Import** コマンドで **Import** ウィンドウを開くと、CD の内容が表示されます。表示されない場合は **Reload CD** ボタンを押してみてください。



Akai ファイルの **Import** ウィンドウ

ウィンドウには CD の中身が、パーティション / ボリューム / プログラム / サンプル別に分類表示されています。ここで選択したサンプルやプログラムを、REAKTOR のマップ・ファイル (*.map) に変換します。変換後のファイルの保存先は、環境設定 (**Directories** ページ) に **Imported Files (Akai)** としてあらかじめ設定しておいたフォルダーになります。

変換の対象になるのは次のようなデータです。

- ・ サンプル・データ
- ・ ループ点
- ・ ルート・キーの設定
- ・ パンの設定
- 一方、次のデータは変換されません。
- ・ フィルター設定
- ・ エンベロープ設定
- ・ ベロシティー・スプリット

- ・ 利得 (ゲイン)

サンプラー系モジュールに直接インポートすることも可能です。
Import ウィンドウで **Load Prg as Map** ボタンを押してください。
すると、サンプル・マップ・エディターの **Edit Sample List > Save Map** コマンドで、サンプル群をマップ・ファイルとして保存できます。あるいは、モジュールの **Store map with Module** プロパティー (**Function** ページ) をオンにすることにより、アンサンプルと組にして保存することもできます。

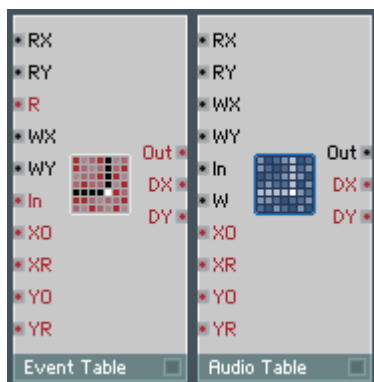
インポート処理を短時間で済ませたい場合は **Mute Audioengine During Conversion** をオンにしてください。但し、処理中は REAKTOR のオーディオ出力がなくなります。

22. テーブル型のモジュール

Audio Table、**Event Table** は、オーディオ・データやイベント・データを操作する自由度が非常に高いモジュールです。まずオーディオ・データについては、波形を (マウスを使って) 自由に設定することにより、発振器 (LFO を含む) や波形シェイパーを作成できます。波形テーブル間のクロスフェードも可能です。さらに、このモジュールを使ったエンベロープ生成器は、エンベロープ曲線を (マウスで) 自由に描画し、ブレイクポイントを好きなだけ設定できます。一方イベント・データについては、ゲート信号やピッチその他のパラメーター変化を自在に制御する、シーケンサーとして使えます。

22.1. プロパティ

Properties ウィンドウで設定できる事項は非常に多岐にわたります。**Event Table**、**Audio Table** とも設定できる項目は同じなのでひとまとめにして解説し、他の一般のモジュールと共通の項目は省略します。



Event Table、*Audio Table* モジュール

Function ページ

Properties

Event Table - Properties

LABEL

Event Table

STATUS

☐ Mono ☐ Mute

INTERPOLATION

None

CLIP/WRAP XY

Wrap

☒ Backup Data With Module

FILE

<empty>

CLIENTS

-

X SIZE **Y SIZE**

empty empty Set

VALUE

MAX	STEPSIZE	DEFAULT
0	0	0
MIN	NUM STEPS	
0	0	

DISPLAY UNITS

X UNITS **Y UNITS**

SAMPLES/SEC	BPM	TICKS/BEAT
0	0.00	0
SAMPLES/TICK	NUM. OF BEATS	
0	0	

Event Table モジュールのプロパティ画面、Function ページ

- ・ **INTERPOLATION:** インデックスとして小数を与えたとき、その前後の整数をインデックスとして表を引き、その値を使って補間した結果を出力することができます。この補間処理を行うかどうかの設定です。
- ・ **None:** 補間を行いません。したがって、インデックス値が連続して変化しても、出力値は階段状に変化することになります。

- ・ **X:** X 軸についてのみ補間を行います。
- ・ **Y:** Y 軸についてのみ補間を行います。
- ・ **XY:** X 軸、Y 軸とも補間を行います。インデックス値が連続して変化すれば、出力値もそれに応じて徐々に変化します。
- ・ **CLIP/WRAP XY:** インデックスとして表にデータが設定されている範囲外の値を与えた場合の動作を設定します。
 - ・ **Clip:** インデックスの最大値 / 最小値よりも大きな / 小さな値を与えた場合、最大値 / 最小値が与えられたものとして処理します。
 - ・ **Wrap:** インデックス範囲の長さで割った剰余を、改めてインデックス値とみなして処理します。
- ・ **Backup Data with Module:** オンにすると、アンサンプル / インストゥルメント / マクロを保存する際、テーブル・データも保存するようになります。
- ・ **FILE:** テーブル・データをファイルから読み込み、またはファイルに保存することができますが、そのファイル名を表します。
- ・ **New** ボタン: 中身が空のテーブルを作ります。
- ・ **Load** ボタン: ファイルからテーブル・データを読み込みます。読み込めるファイルとしては、REAKTOR 独自のテーブル・ファイル (*.ntf)、オーディオ・ファイル (*.wav、*.aif)、テキスト・ファイル (*.txt) があります。テキスト・ファイルの場合は 1 次元のテーブル (**Y SIZE** が 1) に限り、数値を空白で区切って並べた形式で表します。あらかじめ、適当なテキスト・エディターで作っておいてください。
- ・ **Save** ボタン: テーブル・データをファイルに保存します。
- ・ **CLIENTS:** 同じテーブル・データを共有しているモジュールの数。同じファイルを、同じアンサンプルに属する複数のテーブル・モジュールに読み込んだ場合、そのデータは共有されます。したがって、あるテーブル・モジュールで値を変更すると、ほかのモジュールにも影響が及ぶことになります。制御パネルに値を表示していれば、各モジュールに対応するグラフ表示が一斉に変わります。
- ・ **X SIZE、Y SIZE:** テーブルの幅 / 高さ、すなわち、X/Y 軸に沿ったセル数を表します。なお、変更した場合は **Apply** ボタンを押してください。
- ・ **Set:** テーブル・データ用に確保する記憶域の大きさを定義します。

注意: 幅/高さを減らした場合、その分のデータも削除されてしまいます。

- ・ **VALUE: MAX、MIN、STEPSIZE、NUM STEPS** については、ノブやフェーダーと同様です。**DEFAULT** の値は、セルの初期化、テーブルの大きさ拡張、選択部分の切り取りの際、セル値として使います。パネル表示では、デフォルト値になっているセルのみ、ほかとは違う色（通常は黒）で表示されます。初期状態では 0 になっています。
- ・ **DISPLAY UNITS:** マウス操作で値を入力するモードで編集時、グラフのステータス・バーに現在値を表示する単位を表します。
 - ・ **Numeric:** 数値で表示。
 - ・ **MIDI Note:** 小数点以下を四捨五入し、値を MIDI ノート番号とみなして音名で表示します。例えば 60 は「C3」、58 は「A#2」となります。
 - ・ **%:** 値を 100 倍し、「%」記号をつけて表示します。
- ・ **X UNITS:** テーブルの横軸インデックスを指定する単位を表します。
 - ・ **Index:** 0 から始まる番号で指定します（初期状態）。
 - ・ **0...1:** 先頭セルを 0、末尾セルを 1 として線形に補間した、0 ～ 1 の値で表します。
 - ・ **Milliseconds:** ミリ秒単位の値と考え、サンプル・レートに応じて換算して表します。**Audio Table** モジュールで、サンプル・データを格納している場合に有効です。
 - ・ **Tempo Ticks:** ティック単位の値と考え、テンポ設定に応じて換算して表します。**Audio Table** モジュールにリズム・パターンを格納している場合などに有効です。
- ・ **SAMPLES/SEC: X UNITS** が **Milliseconds** の場合に、サンプル・レートを表します。WAV/AIFF ファイルを読み込めば自動的に設定されます。
- ・ **SAMPLES/TICK: X UNITS** が **Tempo Ticks** の場合に、1 ティック当たりのサンプル数を表します。
- ・ **BPM: X UNITS** が **Tempo Ticks** の場合に、1 分当たりの拍数を表します。
- ・ **TICKS/BEAT: X UNITS** が **Tempo Ticks** の場合に、1 拍当たりのティック数を表します。REAKTOR のマスター・クロックを使う場合は 24 になります。
- ・ **NUM. OF BEATS:** 1 拍当たりに相当するセル数を表します。オーディオ・ループを滑らかに再生するためには、4 ～ 8 程度が適当です。





SAMPLES/TICK 以下の 4 つの項目は互いに関連し合っており、データ長やサンプル・レートに合致するよう、1 つを変更すれば他の項目も再計算されます。

- ・ **Y UNITS**: テーブルの縦軸インデックスを指定する単位を表します。
 - ・ **Index**: 0 から始まる番号で指定します (初期状態)。
 - ・ **0...1**: 先頭セルを 0、末尾セルを 1 として線形に補間した、0 ～ 1 の値で表します。

Appearance ページ

Properties ✕

Audio Table - Properties

LABEL

Audio Table

VISIBLE IN VIEW A AND B

☒ Picture ☒ H Scroll Bar
☒ V Scroll Bar

SIZE X
384 Pixels



SIZE Y
128 Pixels

GRAPH
Line

VIEW PARAMETERS

☒ X Auto Fit
☒ Y Auto Fit
☒ Value Auto Fit

ALIGNMENT









GRID

Grid	Step	Sz1	Sz2	Sz3	Sz4
X	-	-	-	-	-
Y	-	-	-	-	-
Value	-	-	-	-	-

ENABLE GRID

☐ X ☐ Value
☐ Y

VISIBLE

☒ Label ☒ Visible
☒ Value ☐ Small Label/Value

Audio Table モジュールのプロパティ画面、Appearance ページ

- ・ **VISIBLE IN VIEW A AND B:** 以下はモジュール全体に関する設定で、A、B 両方の制御パネルに適用されます。
- ・ **Picture:** オンにすると、テーブルがパネル上に表示されるようになります。

- ・ **H Scroll Bar**: オンにすると、テーブル・グラフの下に、左右方向のスクロール・バーが現れます。
- ・ **V Scroll Bar**: オンにすると、テーブル・グラフの右に、上下方向のスクロール・バーが現れます。
- ・ **SIZE X、SIZE Y**: パネルに表示するテーブル領域の大きさを、ピクセル単位で指定します。
- ・ **GRAPH**: 制御パネルでの値の表示方法を、次の 6 つから選択します。
 - ・ **Pixel**: 値を水平線で表示します。X 軸に関して補間を行わない場合、短い横線が、まるで小さなフェーダーが並んでいるように見えます。
 - ・ **Line**: 同様に値を水平線で表示しますが、値が変化するところには垂直線を引いてつなげます。
 - ・ **Bar**: **Line** と同様ですが、線より下の部分を所定の色で塗りつぶします。
 - ・ **2D Color**: X 軸方向だけでなく Y 軸方向の変化も同時に見るための表示方法です。左上隅を原点として、横軸 (X 軸) 方向、縦軸 (Y 軸) 方向に矩形を並べ、色で値を表します。
 - ・ **2D Curve**: **2D Color** と同じく 2 次元テーブルの全体を表示するものですが、値を表す曲面を、小平面に分解し、稜線のみで描画します。**2D Color** はテーブル・データを上から見下ろす形でしたが、**2D Curve** は正面から眺める恰好になります。
 - ・ **Solid**: **Bar** と同様ですが境界線は引きません。
- ・ **VIEW PARAMETERS**
 - ・ **X Auto Fit**: これをオンにすると、X 軸方向の定義域 (セル) 全体が、ちょうどグラフの表示範囲に収まるようになります。セルの個数は **Function** ページで指定した **X SIZE** と一致します。
 - ・ **ALIGNMENT (X)**: **X Auto Fit** がオフの場合は、入力ポート **XO** に与えられた値によって、グラフ領域に表示する X 軸方向の範囲を制御します。**XO** によって決まるセルを、グラフ領域のどの位置に合わせるか、このスライダーで設定します。一番左に合わせた場合、**XO** で指定したセルがグラフ領域の左端になります。中央に合わせると、このセルがグラフ領域の中央になります。一番右にすると、**XO** で指定したセルの直前が、グラフ領域の右端になります。
 - ・ **Y Auto Fit**: グラフの表示形式が **2D Color** または **2D Curve** の場合にのみ有効です。これをオンにすると、Y 軸方向の定義域 (セ

ル) が、ちょうどグラフの表示範囲に収まるようになります。セルの個数は **Function** タブで指定した **Y Size** と一致します。

- **ALIGNMENT (Y):** X軸方向についての **ALIGNMENT** と同様です。
- **Value Auto Fit:** グラフの表示形式が **Piexl**、**Line**、**Bar** の場合にのみ有効です。オンにすると、**Function** ページで指定した **MIN**、**MAX** に合わせて、表示される値の範囲が自動的に決まるようになります。
- **GRID:** 目盛を延長してグラフ面に表示する、目盛線に関する設定です。
 - **ENABLE GRID:** **X** をオンにすると、横軸 (X 軸) の目盛に合わせて縦に目盛線を表示するようになります。同様に、縦軸の目盛に合わせて横に目盛線を表示することもできますが、グラフの表示形式が **2D Color** または **2D Curve** の場合は **Y**、それ以外の場合は **Value** でオン / オフを切り替えます。
 - **Step: X、Y、Value** の各目盛線について、表示間隔を設定します。目盛の 1 単位ごとに線を引くのであれば 1、1 つおきに線を引くのであれば 2 としてください。また、0.5 とすれば、1 単位当たり 2 本ずつ引くことになります。
 - **Sz1、Sz2、Sz3、Sz4:** 目盛線は、最も細い 1 番から最も太い 4 番まで、4 種類の太さを使い分けることができます。例えば **Sz4** に 5 を指定すると、5 本ごとに最も太い線を引く、という意味になります。0 を指定すると、その太さの線をまったく使わないことを表します。
- **VISIBLE: A、B** の各制御パネルに、何を表示し何を表示しないか、を設定します。**AB** について設定すれば、どちらの制御パネルにも共通に適用されます。
 - **Label:** オンにすると、制御パネルにモジュール名が表示されるようになります。
 - **Visible:** グラフ部分を表示し、編集できるようにするか否かを切り替えます。
 - **Value:** オンにすると、グラフの上部にステータス・バーを設け、マウス・カーソル位置の X 座標、Y 座標、およびその値を数字で表示するようになります。値の単位は **DISPLAY UNITS** プロパティー (**Function** ページ) で決まります。
 - **Small Label/Value:** オンにすると、モジュール名や値を小さく表示するようになります。

22.2. コンテキスト・メニュー

テーブル・グラフ上でコンテキスト・メニューを開くと、以下のよう
なコマンドが並んでいます。なお、インストールメントが **Panel Lock**
モードになっていないと、コンテキスト・メニューは開かないので注意
してください。

モード切り替え

- ・ **Table Draw Mode**: マウス操作で値を入力するモードです。マウスで曲線をなぞるようにして入力するか、ある 1 点の値を、マウスを上下することにより増減します。グラフの表示形式が **2D Color** または **2D Curve** の場合は、入力したい値を **Set 2D Draw Value...** であらかじめ指定しておきます。入力したいのと同じ値のセルが他にあれば、**Ctrl** キーを押しながらそのセルをクリックすることによっても値を指定できます。
- ・ **Table Select Mode**: マウスでなぞった領域が選択状態になります。こうしておいて、次にさまざまな編集機能を実行することになります。
- ・ **Table Control Mode**: 値を入力することも、現在の値を変更することもできない状態になります。

ファイル入出力

- ・ **Load Data into Table**、**Save Table Data: Properties** ウィンドウの **Function** ページにある、ファイルの **Load/Save** ボタンと同じ機能です。
- ・ **Save Table Data as...**: 新しい名前のテーブル・ファイルに保存します。
- ・ **Reload Table Data**: REAKTOR 以外でテーブル・ファイルを更新した場合に、もう一度読み込み直します。

表示の切り替え

- ・ **Show All**: テーブル全体が納まるよう、縮小して表示します。
- ・ **Show Selection**: 現在選択されている範囲だけが納まるよう、拡大して表示します。

- ・ **Next Y:** 表示形式が **Pixel**、**Line**、**Bar** の場合は、表示する範囲を表す Y 座標を 1 増やします。**2D Color**、**2D Curve** の場合は 1 列下にスクロールします。
- ・ **Previous Y:** 表示形式が **Pixel**、**Line**、**Bar** の場合は、表示する範囲を表す Y 座標を 1 減らします。**2D Color**、**2D Curve** の場合は 1 列上にスクロールします。

グラフの表示形式

- ・ **Pixel Mode**
- ・ **Line Mode**
- ・ **Bar Mode**
- ・ **2D Color Mode**
- ・ **2D Curve Mode**
- ・ **Solid Mode**

グラフの表示形式を切り替えます。プロパティの設定は変わりません。それぞれの形式について詳しくは、**Appearance** ページの説明を参照してください。

表示

- ・ **Show Read Position:** オンにすると、現在読み込んでいる位置を垂直線で表示ようになります。
- ・ **Show Write Position:** オンにすると、現在書き出している位置を垂直線で表示ようになります。
- ・ **Show Horizontal Position Line:** オンにすると水平方向の目盛が表示されます。
- ・ **Show Horizontal Scroll Bar:** オンにするとグラフの下側に左右方向のスクロール・バーが現れます。
- ・ **Show Vertical Scroll Bar:** オンにするとグラフの右側に上下方向のスクロール・バーが現れます。

選択操作

- ・ **Select All:** 現在画面に現れているデータをすべて選択します。
- ・ **Select X All:** 選択された行に現れているデータをすべて選択します。
- ・ **Select Y All:** 選択された列に現れているデータをすべて選択します。

- ・ **Snap Selection to Grid:** これをオンにすると、選択範囲の境界が常に目盛線 (**Appearance** ページの **GRID**) と一致するようになります。縮小表示すると細い目盛線は見えなくなりますが、この場合は表示されている目盛線にのみ重なります。

データ操作

- ・ **Mirror X:** 選択範囲のデータを左右に反転します。
- ・ **Mirror Y:** 選択範囲のデータを上下に反転します (表示形式が **2D Color**、**2D Curve** の場合)。
- ・ **Rotate/Add/Scale...:** 選択範囲のデータに対してある演算を施します。パラメーターの入力画面が現れるので、数値を入力してください。演算には、パラメーター値の加算 (**Add Value...**)、循環シフト (**Rotation**)、パラメーター値の乗算 (**Scale Value...**) があります。循環シフトの場合、所定の幅だけセルのデータをずらしますが、一方の端から押し出されたセルは反対側の端から再び現れます。
- ・ **Trim Selection:** 選択範囲外のセルを、不要部分として切り取ります。X/Y 軸方向のセル数の設定も更新されます。
- ・ **Delete Rows:** 選択範囲に含まれる行全体を削除します。Y 軸方向のセル数の設定も更新されます。
- ・ **Insert Rows:** 選択範囲に含まれるのと同じ行数分を挿入します。Y 軸方向のセル数の設定も更新されます。
- ・ **Quantize Value to Step Size:** オンにすると、マウス操作で値を入力するとき、**STEPSIZE** プロパティで指定された単位にクオンタイズされるようになります。初期状態ではオンになっています。
- ・ **Set 2D Draw Value...:** 表示形式が **2D Color**、**2D Curve** の場合に、マウスで囲んだ範囲内のセルに対して与える値を設定します。なお、Windows ならば **Ctrl** キーを押しながら右クリック、MacOS X ならば **command** キーを押しながらクリックすることにより、その位置にあるセルと同じ値を設定することができます。

切り取り、コピー、貼り付け

- ・ **Copy:** 選択範囲をクリップボードにコピーします。
- ・ **Cut:** 選択範囲をクリップボードにコピーし、元のテーブルからは削除します。
- ・ **Paste:** クリップボードの内容を貼り付けます。

22.3. その他の機能

編集モード (表示 / 選択) の切り替え

編集モードは、ステータスバーの左端にある正方形の中に表示されます。「D」は表示 (Draw) モード、「S」は選択 (Select) モードを表します。この正方形をクリックするとモードが切り替わります。

また、**tab** キーを押しても編集モードが切り替わります。

セルの移動

セルを選択してマウスでつかみ、**shift** キーを押しながらドラッグすると、この部分を左右に (表示形式が **2D Color**、**2D Curve** ならば上下左右に) 動かせます。全体を選択し、グラフ全体を動かすことも可能です。

セル値の増減

セルを選択してマウスでつかみ、Windows ならば **Ctrl** キー、MacOS X ならば **command** キーを押しながら上下にドラッグすると、各セルの値を増減することができます。

制御パネルのロック

ロック機能に関しては、インストゥルメントとこれに属するテーブル・モジュールとが連動しています。制御パネルのロックがオンになっていると、コピー、貼り付けなどの操作をキーボード・ショートカットで行ったとき、モジュール自体ではなくテーブル内のデータが対象になります。

23. 「Classic Modular」 マクロ・コレクション

「Classic Modular」の名のもとに、往年のアナログ式モジュラー・システムを再現する、高度なマクロを集めました。オリジナルにひけを取らない印象的な音質や適応性に加え、アナログ式では実現できなかった数々の長所を取り入れています。充実したサンプラー、グラニュー・サンプリング・システム、高機能シーケンサーはもとより、発振器、フィルター、音源なども厳選しました。

REAKTOR ライブラリーに独自の位置を占める多くのインストゥルメントが、「Classic Modular」マクロを活用しています。Green Matrix シンセサイザー、Blue Matrix シーケンス・シンセサイザー、Analogic Filter Box など、その響きを耳にするだけで、「Classic Modular」コレクションに秘められた能力を感じ取っていただけるでしょう。

このマクロには次の 2 つの特徴があります。

1. 入出力はすべて、信号の範囲が $-1 \sim 1$ に正規化されています。したがって、変換処理をはさむことなく、ある出力信号をそのまま次のマクロの入力信号として使えるのです。このコレクションに属しないモジュールやマクロは、半音単位、デシベル (dB) 単位、ミリ秒単位などまちまちなので、これと比べても顕著な特徴といえるでしょう。一般には、例えば発振器系モジュールのピッチ (P、音高) は $0 \sim 127$ 、振幅 (A) は $0 \sim 1$ というように、それぞれ範囲が異なるのが普通です。しかしこのコレクションでは、ピッチも $0 \sim 1$ の範囲で制御するようになっています。 $0 \sim 127$ の範囲の信号が必要な場合は、**0-1 to 0-127 Range Converter** マクロ (イベント処理系) を使ってください。

但しシーケンサー系マクロの **Pos** 信号は例外です。整数の番号でステップを表すものなので、 $0 \sim 1$ という範囲にしたのでは意味がありません。

2. 主なパラメーターには、手動で直接制御するつまみ類 (多くはノブ) と、その変調幅を調整するつまみが 1 つか 2 つついています。変調幅の調整つまみには、それに対応した名前の入力端子もあるので、変調源として LFO やエンベロープ・ジェネレーターなどを使えます。入力された変調信号を、変調幅の設定に従って変倍し、手動で指定した値に加算して最終的な値にします。いくつもの変調源が必要な場合は、モジュレーション・ミキサーの出力を変調入力につないでください。こ

の場合、それぞれの変調源の変調幅は最大にしておき、実際の変調幅はミキサー側で調整します。

変調入力にはイベント型とオーディオ型の2種類があります。LFOやエンベロープなど多くの変調源は、どちらの型にも適応できます。しかしオーディオ信号をイベント型の入力として使おうとすれば、その前に**A to E** モジュール (補助系) を置いてイベント信号に変換しなければなりません。

ファイル名に「Event」あるいは「Audio」という文字列が含まれているマクロは、LFOやエンベロープなどの変調源を変倍、ミックス、反転、切り替えするもので、イベント信号用とオーディオ信号用が別々に用意されています。「Stereo」や「Mono」という文字列が含まれていれば、(変調信号ではなく)生のオーディオ信号をミックス、増幅し、ある成分を抑制し(フィルター)、または歪ませるもので、ステレオ用とモノラル用が別になっています。

「Classic Modular」マクロは、いくつかの声部(ボイス)を処理することができます。ほとんどが単声であるアナログ式のモジュラー・システムに比べると、大きな長所であるといえます。インストゥルメントの出力は単声なので、多声信号は**Voice Combiner** モジュール (補助系) で変換する必要があります。なお、メーター、数値表示器、オシロスコープ、ランプなどの表示系マクロはすべて、最後に鳴らされる声部にしか接続できません。

23.1. 表示系マクロ

Number Display (数値表示器)

入力信号の現在値を表示します。値の変化が速すぎる場合は**Peak** スイッチでピーク検出器を有効にするとよいでしょう。振幅のおおまかな変化が包絡線(エンベロープ)の形で表示されます。

Simple Scope (1次元オシロスコープ)

入力信号曲線を、アナログ式オシロスコープと同じように表示します。

XY Scope (2次元オシロスコープ)

XYモードのオシロスコープです。**X**の入力信号が水平軸、**Y**の入力信号が垂直軸に対応します。

23.2. MIDI 処理系マクロ

Controller (MIDI コントローラー)

ピッチベンド、モジュレーション・ホイール、アフタータッチ、ボリュームの各 MIDI コントローラー・メッセージを受信し、そのまま出力します。

Notes - Monophonic (MIDI ノート - 単声)

MIDI ノート・メッセージを受信して、ピッチ、ゲート、ノート・オン・ベロシティの値をそれぞれ該当するポートに出力します。オンになっているノートは同時に 1 つしかない (単声) と考え、すべての声部に同じ信号を出力するようになっています。音域は制御パネルで指定します。**Notes - Polyphonic** マクロと違い、インストゥルメント側で設定される音域とは独立に指定できます。音域設定の異なる別々の **Notes - Monophonic** マクロを同時に使うことも可能です。

Notes - Polyphonic (MIDI ノート - 多声)

MIDI ノート・メッセージを受信して、ピッチ、ゲート、ノート・オン・ベロシティの値をそれぞれ該当するポートに出力します。ノートをいくつもの声部に振り分けて (多声) 出力できます。音域はインストゥルメントのプロパティーとして設定されたものを援用します。したがって、**Notes - Polyphonic** マクロをいくつも使う場合でも、音域はすべて共通です。

Selective Gates - MIDI Keyboard (選択型ゲート - MIDI キーボード)

連続した 12 個のキーを使って、押したキーに対応する出力ポートにゲート信号を振り分けます。キーの範囲は、その最低音を **Lower** ノブで選択することによって指定します。ゲート信号はインストゥルメントのどの声部にも送られます (単声方式)。キーの範囲は、インストゥルメント側とは独立に設定できます。

いくつものエンベロープを用意しておいてキーで切り替える、あるいは演奏中に、使わない音域のキーでシーケンサーの開始 / 停止操作を行う、といった使い方ができます。

Selective Gates - QWERTY - Lower Keys (選択型ゲート - QWERTY キーボードの下段)

コンピューター側のキーボードで所定のキーを押すことにより、外付けMIDIキーボードと同じように、あるインストゥルメントからMIDIノート・イベントを送るようにするマクロです。英語用キーボードのキー配列にちなんで、この機能を「QWERTY」と称します。キーボード下段のZからMまでのキーを押したときに、対応するポートからゲート信号を出力します。ゲート信号はインストゥルメントのどの声部にも送られます(単声方式)。出力ポートはそれぞれのキーに対応して1つずつあります。各キーに対応するMIDIノート番号の範囲は、インストゥルメント側とは独立に設定できます。

いくつかのエンベロープを用意しておいてキーで切り替える、あるいは演奏中に、使わない音域のキーでシーケンサーの開始 / 停止操作を行う、といった使い方ができます。

Selective Gates - QWERTY - Upper Keys (選択型ゲート - QWERTY キーボードの上段)

Selective Gates - QWERTY - Lower Keysと同様ですが、キーボード上段の、QからPまでを使います。

23.3. ミキサー / アンプ系マクロ

アンプ (増幅器) について

発振器 (オシレーター) やサンプラーなどの音源には、初めからアンプ (増幅器) が組み込まれています (**A** 入力)。アンプ系マクロは、それ以外の信号を増幅 / 減衰させたい場合に使ってください。

変調前のオーディオ信号を増幅する方法は、このあとどのような信号で変調するか、その信号特性によって2通りあります。人間は振幅の変化を指数函数的に感じているので、振幅を変える変調信号かまたはアンプ自身のいずれか一方に、指数函数的な特性を持たせる必要があります。例えばADSRエンベロープを見ると、デケイ時やリリース時には指数函数的に変化しているので、音源に組み込みのものと同一、線形の特性をもつアンプ (線形増幅器) を接続すればよいことになります。一方、

モジュレーション・シーケンサーから得られる変調信号は線形の特性を持つので、指数増幅器を接続しなければなりません。

変調信号の増幅には線形増幅器が適しています。

Amp - Exponential (アンプ - 指数型の特性)

指数函数的な特性をもったアンプです。振幅はデシベル (dB) 単位で設定します。

Amp - Linear (アンプ - 線形の特性)

線形の特性をもったアンプです。振幅は0～1の範囲で設定できます。

Crossfade (クロスフェード)

クロスフェード・モジュールです。2つの入力信号をミックスして出力します。

Inverter (インバーター)

信号の極性を反転します。単極 (Uni) モードの場合、レベルが0.5の軸を中心として反転します。ベロシティによって影響されないゲート信号に向いています。一方、双極 (Bi) モードにすると、-1を掛ける形で、0の軸を中心として反転します。LFOなど双極の変調源を反転する場合に向いています。

Master Volume (マスター・ボリューム)

マスター・レベルを制御するマクロです。クリッピングを避けるため、リミッターが組み込まれています。インストゥルメントを構築する場合は、その最終段に置かなければなりません。マクロの入力信号は単声なので、多声の信号の場合は **Voice Combiner** モジュール (補助系) で変換してください。

Mixer - Simple (ミキサー - 単純型)

オーディオ信号の単純な4チャンネル・ミキサーです。なお、変調信号のミックスには **Modulation Mixer** マクロを使ってください。

Mixer - Studio (ミキサー - スタジオ型)

8チャンネルのミキサーです。フェーダーの前段および後段にエフェクト・センドがあり、さらにパン機能と、各チャンネルのオン / オフ・ボタンがついています。マクロの入力信号は単声なので、多声の信号の場合は Auxiliary フォルダーにある VC(ボイス・コンバイナー) モジュールで変換してください。

Modulation Mixer (モジュレーション・ミキサー)

変調信号用の 3 チャンネル・ミキサーです。入力信号の反転スイッチがついています。ノブの値が小さい場合にも高い分解能が必要ならば、**exp** ボタンでノブを指数特性に切り替えるとよいでしょう。

変調幅をミキサーで調整する場合、これに接続された各マクロは、変調幅が最大になるように設定してください。

Modulation Matrix - Mixer (モジュレーション・マトリクス - ミキサー)

変調信号用の 8×8 ミキサーです。各列が入力ポート、行が出力ポートに対応します。すなわち、それぞれの行で、どの変調信号をミックスして出力するか、組み合わせを設定するようになっています。ノブの値が小さい場合にも高い分解能が必要ならば、**exp** ボタンでノブを指数特性に切り替えるとよいでしょう。

変調幅をミキサーで調整する場合、これに接続された各マクロは、変調幅が最大になるように設定してください。

Modulation Matrix - Switch (モジュレーション・マトリクス - スイッチ)

変調信号用の 8×8 スイッチです。各列が入力ポート、行が出力ポートに対応します。すなわち、それぞれの行で、どの変調信号を出力するか設定するようになっています。変倍その他の処理は行わず、信号をそのまま出力します。

Panner (パン)

入力信号を左右 2 つの出力に振り分けて定位を調整します。

Scanner (スキャナー)

スキャン位置を表す入力値に応じて、8つの入力の中から信号を選択します。スキャン位置が整数でない(2つの入力の間)場合は、その小数部分の大きさに応じて按分(クロスフェード)して出力します。

23.4. 発振器系マクロ

Geiger Counter (ガイガー・カウンター)

衝撃波やパルス波をランダムな間隔で生成し、ガイガー・カウンター(放射能レベル測定器)のような音を出します。衝撃波やパルス波を生成する平均頻度およびランダム性の度合いを設定できます。

Noise (ノイズ ; 雑音)

4種類の雑音を生成できます。

白色雑音: あらゆる周波数成分をほぼ同量含む雑音です。

桃色雑音: 周波数に反比例(オクターブ当たり3dBの割合で減少)して、高い周波数成分ほど弱くなる雑音です。バンド・パス・フィルター付きのフィルター・スweepを通すと振幅が一定になります。

有色雑音: 「色」、すなわち特に振幅の大きい周波数帯域は、**Color** ノブで調整できます。

808: 伝説のドラム・マシンであるTR-808を模した、ハイハットやシンバルの音を合成します。調律を狂わせたパルス発振器が6つ組み込まれています。

Oscillator - Symmetry (発振器 - 対称型)

対称 / パルス幅変調機能のついた発振器です。発振波形としては、双極ランプ・パルス波(鋸波に似た波形)、双極パルス波、一般的なパルス波、三角波 / 鋸波、放物波(正弦波に似た波形)などが可能です。変調入力には、特性の異なる2種類があります。一方は線形で、LFOの出力信号を変調するのに向いており、もう一方は指数的で、エンベロープの変調に向いています。

Oscillator - Sync (発振器 - 同期型)

多用途の発振器で、ハード / ソフト同期、位相変調 / 周波数変調の機能が組み込まれています。発振波形としては、鋸波、パルス波、三角波、正弦波、衝撃波があります。ハード同期をかけると、所定の位相に到達した時点で再び位相が 0 に戻るようになります。ソフト同期の場合は、所定の位相に達すると、位相を反転して (逆向きに) 戻る形の波形になるため、より「穏やかな」同期音になります。

Random (ランダム発振器)

レベルがランダムに変化するサンプル & ホールド方式の発振器。所定の頻度で乱数を発生してレベルを決め、次に乱数を求めるまでの時間はそのレベルを保持します。**Rmp** スイッチをオンにすると、次のレベルに一瞬で変わるのではなく、ある程度の傾斜をもって変化するようになります。

23.5. サンプラー系マクロ

サンプラーはサンプルを再生します。再生するサンプルは、サンプル・マップに登録しておく必要があります。**Properties** ウィンドウの **gearwheel** ページに、サンプル・マップを編集するためのエディターがあります。**Properties** ウィンドウは、サンプル表示部分をダブル・クリックして開いてください。サンプラーについている **Select** ノブで、再生するサンプルを切り替えます。

ビート・ループ・サンプルを再生するためには、**Classic Sampler** または **Beat Loop** を使います。**Beat Loop** サンプラーには、テンポとピッチを独立に設定できるという長所がありますが、その必要がないのであれば **Classic Sampler** を使ってください。ループ長も、他の主なパラメーターと同様に、変調の対象とすることができます。4 分音符相当、1 小節相当など、音楽的に切りのよいループ長にするため、**LL Q** ノブで量子化 (クオンタイズ) の設定をしておくといでしょう。これが 0 であれば量子化は起こりません。**Beat Loop** の場合は、サンプルの開始位置、ループの開始位置およびオフセットについても、同様に量子化の設定が可能です。なお、**Classic Sampler** の場合はループのオン / オフをプロパティーで設定できますが、**Beat Loop** は常にループ・モードに固定です。

Beat Loop (ビート・ループ)

ビート・ループの再生に特化したサンプラーです。

どのようなビート・ループ・サンプルでも、元のテンポとは無関係に、**Clk** 入力に与えられるクロック源に同期させることができます。再生ピッチはテンポと独立に設定できます。サンプルは無限に繰り返して再生されます。

サンプル選択や再生ピッチばかりでなく、再生開始位置やループ開始位置、ループ長、オフセット位置も設定、変調することができます。それぞれの位置の調整つまみは量子化 (クオンタイズ) が可能で、その単位 (16 分音符の倍数) は **Unit** ノブで設定します。

現在の再生位置を通知する、**Pos** および **Pos*** というデータ出力があって、シーケンサーの駆動用に使えます。**Pos** は 16 分音符相当の長さごとに位置イベントを出力します。但しこれは、**Beat Loop** 自身のパラメーターを変調するためには使えません。もう一方の **Pos*** 出力からは、「グレイン」ごとに位置イベントが生成されます。これは次の 16 分音符よりも 32 分音符相当分だけ前の位置 (裏拍の位置) に当たり、開始位置以外のパラメーターはすべて、これに合わせてサンプリングされるようになっています。したがって、シーケンサー側でいずれかのパラメーターを変調するのであれば、これに **Pos*** 信号を与えて同期させる必要があります。

Classic Sampler (クラシック・サンプラー)

従来型のサンプル再生方式なので、テンポとピッチが独立にはなりません。周波数が変調可能であること、変調信号で再生速度を制御し、逆方向再生も可能であることが特徴です。ループのオン / オフは、サンプルごとに独立に、サンプル・マップ・エディターで設定できます。

開始位置、ループ開始位置、ループ長は、サンプル長の 128 分の 1 を単位として設定できます。例えば 1 小節分の長さのサンプルであれば、8 単位が 16 分音符 1 つ分に相当します。

Resynth (リシンズ ; 分割 - 再合成方式のサンプラー)

ピッチとテンポを独立に制御できるサンプル・リシンセサイザーで、ビートなしのサンプル再生に特化されています。サンプルを「グレイン」という小断片に分割し、(順序を入れ替えて) 1 つずつ再生します。単位

時間あたりのグレイン数を、**Granu** ノブで設定できます。グレイン間の境界でレベルが急激に変化するのを防ぐため、隣り合うグレインが部分的に重なるようにすることもできます。この場合の重複時間は **Smooth** ノブで設定します。

開始位置、ループ開始位置、ループ長は、サンプル長の 128 分の 1 を単位として設定できます。例えば 1 小節分の長さのサンプルであれば、8 単位が 16 分音符 1 つ分に相当します。

23.6. シーケンサー系マクロ

高性能シーケンサーの構築に必要なマクロを集めてあります。狭義のシーケンサーのほか、関連機能として、クロック信号の生成 / 変調マクロがあります。シーケンサーはクロック信号に同期して動作しますが、生成マクロはこの信号を、所定の頻度で、値を 1 ずつ増やしながらイベントとして生成します。この値と、シーケンサーの処理対象であるシーケンス上のある位置とが対応しているので、このイベントを「位置イベント」といいます。

変調マクロは一連の位置イベントに変化を加えるもので、生成マクロとシーケンサーの間に挿入して使います。

Global Clock (グローバル・クロック)

位置イベント (クロック) を生成、送出するマクロです。クロックを開始 / 停止するボタンが REAKTOR のツールバーにあります。生成する頻度はコントロール・パネルで設定できます。また、クロック信号とは別に、停止中は 0、動作中は 1 となるゲート信号を出力します。シーケンサーによっては初期化のためにゲート信号を必要とするほか、ノート・シーケンサーの停止時にノート・イベントを正しく制御するためにも必要です。

Position Delay (位置イベントのディレイ)

位置イベント (クロック信号) にランダムな遅延を加え、間隔が一定でないようにするマクロです。

Position Looper (位置イベントのループ)

位置イベントをループ化するマクロです。シーケンサー・マクロに組み込みのループ機能と違い、ループの開始位置や長さを変調することも

できます。ループ末尾に到達すると再び開始位置に戻るわけですが、その間に開始位置が変わって (変調されて) いた場合の動作によって、ハード同期 (Hardsync)、自走 (Freerun) という2つのモードがあります。ハード同期モードであれば、直ちに新しい開始位置に戻ります。通常はこのモードで使うとよいでしょう。自走モードの場合は、変わる前の開始位置にいったん戻り、そこから新しい開始位置に進んで新しいループに入るので、このずれに相当する分の時間が余分にかかります。

Position Offset (位置イベントのオフセット)

位置イベントの値にオフセットを加えることにより、シーケンス上の位置をシフトしたのと同じ効果を実現します。

Sequencer - 1x Notes, 4x Mod, 8x Trigger (シーケンサー - ノート、4トラック・モジュレーション、8トラック・トリガー)

ノート・シーケンサー、4トラックのモジュレーション・シーケンサー、8トラックのトリガー・シーケンサーを組み合わせたものです。同じシーケンス番号で制御するほか、ループ・バー、編集バー、ビュー・バーなどの制御つまみや、コピー、貼り付け、クリア、録音開始 / 停止などの機能も共通になっています。詳しくは **Sequencer - Note**、**Sequencer - Modulation 4x**、**Sequencer - Trigger 8x** の各マクロに関する項を参照してください。なお、REAKTOR ライブラリーの Blue Matrix シンセサイザーはこのマクロで制御されています。これについては Blue Matrix の資料を参照してください。

Sequencer - Classic Step (シーケンサー - クラシック・ステップ)

16 ステップの伝統的なシーケンサーです。このコレクションに収められた他のシーケンサーと違って、ステップごとに別々のフェーダーで調整できるようになっています。ステップごとの値は、フェーダー・ボックスなどの MIDI コントローラーで遠隔制御することも可能です。

Sequencer - Modulation 4x (シーケンサー - 4トラックの変調信号用)

4系統の変調信号を独立に生成できるシーケンサーです。発振器やフィルターのパラメーターを変調するために使います。LFO やエンベロープ生成器と同じ役割と考えればよいでしょう。

View ボタンを押すと、**all** と **solo** が交互に切り替わります。**all** ビューの場合、すべてのチャンネルが同時に表示されます。**solo** ビューにすると1チャンネル分の表示になりますが、上下方向の分解能が上がり、より詳しく見ることができます。左から2つ目にある **Select** バーで、どのチャンネルを表示するか指定します。

128 種類のシーケンスから **Seq** ノブで選択できるようにするため、各スナップショットに一意的なシーケンス番号を割り当ててください。シーケンス長は 768 ステップで、分解能が 96 分音符相当ならば 8 小節分、16 分音符相当ならば 48 小節分に当たります。シーケンスの番号と長さは **Properties** ウィンドウの **gearwheel** ページで設定します。**X** が長さ、**Y** がシーケンス番号です。コピー / 貼り付け操作および取り消し操作に最後の 8 シーケンス (4 チャンネル× 2) 分を使うので、少なくとも 12 シーケンス (4 チャンネル× 3) は必要になります。

このシーケンサーには編集バッファがありません。何か変更すれば直ちに確定し、保存されてしまうので注意してください。元のシーケンスを残したままで多少変更したシーケンスを作りたい場合は、コピー / 貼り付け機能を使って複製を作り、そちらを編集するようにしなければなりません。

Properties ウィンドウの **Eye** ページには時間軸に沿った格子が表示されており、クロック源から **Pos** 入力ポートに与えられる位置イベントを基準に、シーケンスを調整できるようになっています。位置イベントが 96 分音符相当の分解能ならば **Grid step** を 6 に、16 分音符相当ならば 1 にしてください。時間軸の目盛は **Grid step** の設定に応じて変わります。

画面上には編集バー、ループ・バー、ビュー・バーの 3 種類の水平バーがあります。この大きさは、左右の端をつかんでドラッグすれば調整できます。端以外の部分の動作はスクロール・バーと同様であり、バー本体の左側または右側をクリックすれば 1 単位分スクロールすることも同じです。編集バーはシーケンス上の選択範囲を調整するもので、コピー

/ 貼り付け / 切り取り / 挿入、ランダム化、量子化 (クオンタイズ)、クリア、録音の各コマンドに適用されます。ループ・バーを調節してシーケンス上の範囲を選択すると、その部分がループ再生されます。ビュー・バーはシーケンスのうち画面に表示する範囲を調整します。なお、編集バーとループ・バーについては、**Bar /** つまみで量子化の設定ができます。

各ボタンの上にマウス・ポインターを持っていくと、簡単な説明が現れます。なお、ボタンを押すと、画面上に見えているチャンネルすべてに対してその機能が適用されます。

Mod ポートに入力される変調信号は、各チャンネルの **recE** ボタンをオンにすると、そのまま録音できるようになります。**Rec 1/0** ボタンを押すと録音が始まります。なお、録音できるのは編集バーで設定された範囲に限ります。**1 shot** ボタンをオンにしておくと、**Rec 1/0** ボタンを押した後、最初の変調イベントを受け取った時点で録音が始まるようになります。編集バーで設定された範囲を過ぎると、自動的に録音は停止します。

Sequencer - Note (シーケンサー - ノート)

標準的なピアノ・ロール方式で、ノート・イベントを並べて定義したシーケンスに沿って信号を生成するシーケンサーです。画面は上下2つの領域に分かれています。上側にはノート・イベントがピアノ・ロール方式で表示されています。すなわち、水平方向が時間軸、垂直方向が音高 (ピッチ) を表します。ピッチ・グラフが所定の閾値を越えるとノートの再生が始まり、閾値を下回ると停まります。この閾値は、上のデータ欄にある2つ目の垂直バーで調整できます。画面下側は、各ノートの再トリガー・イベントを生成し、ベロシティを設定するために使います。

128 種類のシーケンスから **Seq** ノブで選択できるようにするため、各スナップショットに一意的なシーケンス番号を割り当ててください。シーケンス長は 768 ステップで、分解能が 96 分音符相当ならば 8 小節分、16 分音符相当ならば 48 小節分に当たります。シーケンスの番号と長さは **Properties** ウィンドウの **gearwheel** ページで設定します。**X** が長さ、**Y** がシーケンス番号です。コピー / 貼り付け操作および取り消し操作に最後の 2 シーケンス分を使うので、少なくとも 3 シーケンスは必要になります。

このシーケンサーには編集バッファがありません。何か変更すれば直ちに確定し、保存されてしまうので注意してください。元のシーケンスを残したままで多少変更したシーケンスを作りたい場合は、コピー / 貼り付け機能を使って複製を作り、そちらを編集するようにしなければなりません。

Properties ウィンドウの **Eye** ページには時間軸に沿った格子が表示されており、クロック源から **Pos** 入力ポートに与えられる位置イベントを基準に、シーケンスを調整できるようになっています。位置イベントが 96 分音符相当の分解能ならば **Grid step** を 6 に、16 分音符相当ならば 1 にしてください。時間軸の目盛は **Grid step** の設定に応じて変わります。

画面上には編集バー、ループ・バー、ビュー・バーの 3 種類の水平バーがあります。この大きさは、左右の端をつかんでドラッグすれば調整できます。端以外の部分の動作はスクロール・バーと同様であり、バー本体の左側または右側をクリックすれば 1 単位分スクロールすることも同じです。編集バーはシーケンス上の選択範囲を調整するもので、コピー / 貼り付け / 切り取り / 挿入、ランダム化、量子化 (クオンタイズ)、クリア、録音の各コマンドに適用されます。ループ・バーを調節してシーケンス上の範囲を選択すると、その部分がループ再生されます。ビュー・バーはシーケンスのうち画面に表示する範囲を調整します。なお、編集バーとループ・バーについては、**Bar / つまみ** で量子化の設定ができます。

各ボタンの上にマウス・ポインターを持ていくと、簡単な説明が現れます。

P および **G** ポートに入力されるピッチ信号、ゲート信号は、**recE** ボタンをオンにすると、そのまま録音できるようになります。**Rec 1/0** ボタンを押すと録音が始まります。なお、録音できるのは編集バーで設定された範囲に限ります。**1 shot** ボタンをオンにしておくと、**Rec 1/0** ボタンを押した後、最初の変調イベントを受け取った時点で録音が始まるようになります。編集バーで設定された範囲を過ぎると、自動的に録音は停止します。

Sequencer - Simple Modulation (シーケンサー - 簡易型、変調信号用)

Sequencer - Modulation 4x と同様のシーケンサーですが、1 チャネル分に限定して簡潔化したものです。

Sequencer - Trigger 8x (シーケンサー - 8トラック・トリガー)

8系統のトリガー信号を独立に生成できるシーケンサーです。エンベロープ、サンプラー、ドラム・シンセサイザーなどのトリガー信号として使えます。

View ボタンを押すと、**all** と **solo** が交互に切り替わります。**all** ビューの場合、すべてのチャンネルが同時に表示されます。**solo** ビューにすると1チャンネル分の表示になりますが、上下方向の分解能が上がり、より詳しく見るができます。左から2つ目にある **Select** バーで、どのチャンネルを表示するか指定します。

128種類のシーケンスから **Seq** ノブで選択できるようにするため、各スナップショットに一意的なシーケンス番号を割り当ててください。シーケンス長は768ステップで、分解能が96分音符相当ならば8小節分、16分音符相当ならば48小節分に当たります。シーケンスの番号と長さは **Properties** ウィンドウの **gearwheel** ページで設定します。**X** が長さ、**Y** がシーケンス番号です。コピー / 貼り付け操作および取り消し操作に最後の16シーケンス(8チャンネル×2)分を使うので、少なくとも24シーケンス(8チャンネル×3)は必要になります。

このシーケンサーには編集バッファがありません。何か変更すれば直ちに確定し、保存されてしまうので注意してください。元のシーケンスを残したままで多少変更したシーケンスを作りたい場合は、コピー / 貼り付け機能を使って複製を作り、そちらを編集するようにしなければなりません。

Properties ウィンドウの **Eye** ページには時間軸に沿った格子が表示されており、クロック源から **Pos** 入力ポートに与えられる位置イベントを基準に、シーケンスを調整できるようになっています。位置イベントが96分音符相当の分解能ならば **Grid step** を6に、16分音符相当ならば1にしてください。時間軸の目盛は **Grid step** の設定に応じて変わります。

画面上には編集バー、ループ・バー、ビュー・バーの3種類の水平バーがあります。この大きさは、左右の端をつかんでドラッグすれば調整できます。端以外の部分の動作はスクロール・バーと同様であり、バー本体の左側または右側をクリックすれば1単位分スクロールすることも同じです。編集バーはシーケンス上の選択範囲を調整するもので、コピー / 貼り付け / 切り取り / 挿入、ランダム化、量子化(クオンタイズ)、クリア、録音の各コマンドに適用されます。ループ・バーを調節してシー

ケンス上の範囲を選択すると、その部分がループ再生されます。ビュー・バーはシーケンスのうち画面に表示する範囲を調整します。なお、編集バーとループ・バーについては、**Bar /** つまみで量子化の設定ができます。

各ボタンの上にマウス・ポインターを持っていくと、簡単な説明が現れます。なお、ボタンを押すと、画面上に見えているチャンネルすべてに対してその機能が適用されます。

Trig ポートに入力されるトリガー信号は、各チャンネルの **recE** ボタンをオンにすると、そのまま録音できるようになります。**Rec 1/0** ボタンを押すと録音が始まります。なお、録音できるのは編集バーで設定された範囲に限ります。**1 shot** ボタンをオンにしておくと、**Rec 1/0** ボタンを押した後、最初の変調イベントを受け取った時点で録音が始まるようになります。編集バーで設定された範囲を過ぎると、自動的に録音は停止します。

23.7. LFO、エンベロープ系マクロ

Envelope - ADSR (エンベロープ - ADSR)

アタック - デケイ - サスティン - リリースのパラメーターを備えた伝統的なエンベロープ生成器。

Envelope - Decay (エンベロープ - デケイ)

デケイのパラメーターを備えたエンベロープ生成器。

Envelope - One-Ramp (エンベロープ - ランプ型)

所定の始点から終点にかけて、徐々にレベルが上昇するエンベロープを生成します。**lin** ボタンで、線形に上昇するか指数函数的に上昇するかを切り替えることができます。**s/h** ボタンをオンにすると、トリガーがかかった時点で、始点と終点の値をサンプル & ホールドするようになります。

Envelope Follower (エンベロープ・フォロワー)

入力信号の振幅に応じたエンベロープを生成します。

Peak モードにすると、入力信号のピーク振幅に応じたエンベロープになります。リリース時間の調整により、入力信号の変化を滑らかにすることができます。アタック時間は常に 0 です。

Roots means square (Rms) モードにすると、入力信号の音量に応じたエンベロープになります。瞬間的に入力レベルが上がっても音量はそれほど変わらないため、**Peak** モードと違って、エンベロープにはあまり影響がありません。技術的にいうと RMS 値は、レベル値の 2 乗を所定の時間間隔にわたって平均し、その平方根を取って求めます。

LFO (LFO、低周波発振器)

低周波発振器。変化の緩やかなランダム波、正弦波、三角波、パルス波を生成できます。波形は **Wave** つまみで切り替えますが、隣り合う波形の間も連続的に変化し、自由に設定できるようになっています。波形の対称性 (パルス幅) は **Width** つまみで調整します。上昇方向に傾斜のある鋸波は、三角波の **Width** を 1 にすれば実現できます。同様に、-1 とすれば、下降方向に傾斜のある鋸波になります。

LFO の発振周波数を設定する方法は 3 通りあり、マクロの左上隅にある **Unit** スイッチでモードを切り替えます。**P** モードにすると、**Speed** つまみを使って、半音単位でピッチを調整できます。**bpm** モードならば、REAKTOR のグローバル・クロックによって決まる、分あたりの拍数が基準になります。**pos** モードも同様ですが、**Pos** ポートに入力されるイベント周波数が基準となります。グローバル・クロック以外のクロックに同期させたい場合は、**pos** モードにし、基準クロックを **Pos** ポートに与えてください。次に、グローバル・クロックとは独立に、**Pos** ポートに与えられるイベントの速度を基準として、拍数を数える単位を **Div** つまみで設定できます。**bpm** モードならば、**Speed** つまみによる調整単位を、96 分音符の倍数として指定します。例えば 6 ならば 16 分音符、12 ならば 8 分音符、24 ならば 4 分音符に相当します。**pos** モードの場合も、入力される位置イベントの分解能が 96 分音符相当であれば同じ割合になります。

外部から与えられる信号に同期させるほか、**Unit** が **bpm** または **pos** の場合、ソング位置に同期させることも可能です。**Phase** つまみで、同期が起こった直後の出力信号の位相を設定してください。

Sample and Hold (サンプル & ホールド)

TE ポートに与えられるイベント、または **C** 入力のクロック・エッジで入力信号の値をサンプリングし、次のイベント / エッジまでその値を保持します。サンプル取得に関する条件はパネルで設定できます。

Triggered Random (トリガーつき乱数生成器)

入力信号のクロック・エッジごとに、乱数を生成して送信します。**Edge** スイッチで、どのエッジでトリガーがかかるかを指定できます。**+** ならば 0 からの立ち上がり、**-** ならば 0 からの立ち下がり、**both** ならばその両方になります。

23.8. フィルター系マクロ

3 Band Filter (3 帯域フィルター)

多用途の 3 帯域イコライザーです。帯域ごとに減衰レベルを調整することにより、ハイ・パス、バンド・パス、ロー・パス、ノッチの各フィルターを実現できます。

ディスク・ジョッキー用ミキサーの「キル」フィルターを模倣することもできます。

Bandsplit (帯域分割)

入力信号を高、中、低の 3 つの帯域に分割します。この 3 つの出力をミックスすれば、元の信号と同じになるはずですが。

Comb (コム・フィルター)

フランジャーやコーラスの効果を作り出すフィルターです。入力信号に、遅延を加えた信号を重ね合わせることで、周波数軸に沿っていくつもの山と谷を繰り返す「櫛」型の周波数特性が得られます。遅延時間の逆数をフィルター周波数とすると、その 0.5 倍、1.5 倍、2.5 倍などの周波数で元の信号と遅延信号とが打ち消しあい、谷が現れます。

遅延信号にフィードバックをかける割合 (レゾナンス) も設定できます。**GainC** ノブは、フィードバック量を増やすと出力レベルが下がる割合を調整するもので、フィードバックを変調する場合には有用です。**Ex**

スイッチにより、フィードバック経路をマクロ外に取り出すことができます。すなわち、**Send** ポートに出力される遅延信号をマクロ外で変調し、**Ret** ポートに返してやるのです。この経路の信号が大きくなりすぎないように、変調器の増幅度は1以下にしておかなければなりません。また、この遅延信号にフィルターをかける場合は **Multimode - Accurate** マクロを使い、**GainC** を1とすればよいでしょう。

Ladder Lowpass (ラダー型ロー・パス・フィルター)

Robert Moog 氏が考案したラダー型フィルター回路をもとに設計したロー・パス・フィルターです。滑らかに飽和するレゾナンス特性、高レゾナンスにしたときの自己発振、周波数変調 (FM) などの特徴があります。

1 ～ 4 ポールの 4 種類のロー・パス・フィルターを組み合わせたものです。「ポール」とは、カット・オフ周波数より高い信号の減衰率を表す数字で、オクターブあたり 6dB の減衰率ならば 1 ポール、24dB ならば 4 ポールとなります。**Poles** ノブの切り替えにより、4 つのフィルター出力のうちどれを最終的に出力するかが決まります。目盛と目盛の間に合わせた場合は、隣り合う 2 つのフィルター出力を、ノブの位置に応じて按分します。

Multimode - Accurate (マルチモード - アキュレート)

CPU への負荷が小さなマルチモード・フィルターです。周波数の違いを正確に認識して処理できるのが特徴で、意図しない周波数の信号を増幅 / 減衰してしてしまうことはほとんどありません。**GainC** ノブは、フィードバック量を増やすと出力レベルが下がる割合を調整するもので、フィードバックを変調する場合には有効です。**GainC** を1とすると、レゾナンスにかかわらず、どの周波数帯域でも増幅度が1以下になります。ディレイ・ラインやコム・フィルターのフィードバック経路にこのマクロをはさむ場合は、このように設定してください。増幅度が1より大きいと、フィードバック・ループの信号がどんどん大きくなってしまいうからです。

次のような種類のフィルターとして使うことができます。

- ・ 減衰率が 6dB/oct. のロー / ハイ・パス・フィルター
- ・ 減衰率が 12dB/oct. のロー / バンド / ハイ・パス・フィルター
- ・ 減衰率が 24dB/oct. のロー / バンド / ハイ・パス・フィルター

Multimode - Resonance Limiter (マルチモード - レゾナンス・リミッター)

レゾナンス・リミッター内蔵のマルチモード・フィルターです。**limit**つまみで、制限する閾値を dB 単位で指定します。バンド・パス・フィルターの信号がこのレベルを超えると、どのフィルターについても、レゾナンスが抑制されます。突出した周波数成分が大きくなりすぎるのを防ぐ働きがあります。また、カット・オフ周波数を増やしたときに閾値を下げる比率を、**F foll** つまみで指定できます。

GainC ノブは、フィードバック量を増やすと出力レベルが下がる割合を調整するもので、フィードバックを変調する場合には有用です。

次のような種類のフィルターとして使うことができます。

- ・ 減衰率が 12dB/oct. のロー / バンド / ハイ・パス・フィルター
- ・ 減衰率が 24dB/oct. のロー / バンド / ハイ・パス・フィルター

23.9. ディレイ系マクロ

Delay (ディレイ)

入力信号を、**Delay** ノブで設定された時間遅らせて出力します。

Delay ノブの目盛 (単位) は 3 通りあり、**Mode** スイッチで選べます。**ms** モードならばミリ秒単位です。**bpm** モードにすると、REAKTOR のグローバル・クロックによって決まる分あたりの拍数が基準になります。具体的な単位は **Unit** つまみで指定し、例えば **1/16** とすれば 16 分音符が単位になります。**pos** モードも同様ですが、**Pos** ポートに入力されるイベント周波数が基準となります。このように、**Delay** ノブの 1 目盛は、基準となる時間と **Unit** つまみの値によって決まります。例えば与えられる位置イベントに 96 分音符の分解能があれば、**Unit** を 6 とすると 16 分音符に相当する長さが 1 目盛になります。グローバル・クロック以外のクロックに同期させたい場合は、**pos** モードにし、基準クロックを **Pos** ポートに与えてください。

遅延時間を変調することも可能です。変調信号を量子化 (クオンタイズ) する場合は、**Unit** つまみで指定された単位を基準として、**MQ** つまみで指定してください。

遅延のかかった信号をフィードバックする割合も調整できます。**Ex** スイッチにより、フィードバック経路をマクロ外に取り出すことができ

ます。すなわち、**Send** ポートに出力される遅延信号をマクロ外で変調し、**Ret** ポートに返してやるのです。この経路の信号が大きくなりすぎないように、変調器の増幅度は1以下にしておかなければなりません。また、この遅延信号にフィルターをかける場合は **Multimode - Accurate** マクロを使い、**GainC** を1とすればよいでしょう。

Dry/Wet は、元の信号と遅延のかかった信号を混ぜ合わせる比率を設定します。

23.10. 歪み系マクロ

入力信号に歪みを与える、クリッパーやサチュレーターなどのマクロには、入力信号レベルを 0dB に揃えるため、入力段に増幅器が組み込まれています。レベル・メーターが橙色にならない場合は、増幅器を働かせるようお勧めします。つまみ類の調整範囲もこれを前提として設計されています。

歪み系マクロに共通するノブとしては、利得を補正する **GainC** もあります。機能はすべて同じですが、ここではクリッパーの場合について説明しましょう。値を0とすると、利得の補正は起こりません。したがって、クリッピング・レベルが低くなれば出力レベルも下がってしまいます。ディレイ系マクロのフィードバック経路に挿入する場合は、増幅の必要がないのでこのように設定してください。**GainC** を1とすれば、失われた分の音量を補償するようになります。クリッピング・レベルによらず、入力信号レベルが 0dB ならば出力も 0dB になるのです。クリッピング・レベルを変調するような場合に特に便利な機能です。但し、入力信号が 0dB に調整されていなければ、想定どおりには働きません。

対称性を調整する **Sym** ノブも多くのマクロについています。信号が正の部分と負の部分とで、歪みを加える程度を違えるかどうかを調整します。0 とすればどちらも同じ程度の歪みを加えるようになります。

Clipper (クリッパー)

入力信号が所定のレベルを超えた部分をクリップします。

Quantizer (クオンタイザー)

入力信号の振幅を量子化 (クオンタイズ) します。階段波の形に歪むので、分解能の低いサンプラーの動作を模擬することになります。

Ringmodulator (リング変調器)

振幅変調、またはその特別な場合であるリング変調を施します。**Mod Depth** つまみで入力信号の振幅を変調する度合いを設定しますが、これを 1 にすればリング変調になります。入力信号に変調信号を掛け合わせることで、新たな倍音成分を加えた信号が得られます。

Saturator (サチュレーター)

チューブ・ディストーションやテープ・サチュレーションの効果を模擬する、オーバードライブ型のモディファイヤーで、特性曲線が滑らかになっています。

Slew Limiter (スルー・リミッター)

信号レベル急変時にその勾配を緩和する (応答速度を下げる) フィルターです。

あるレベルを超えた振幅変化を抑えるリミッターですが、波形の上昇部分と下降部分とで異なるレートを適用できます。変調信号を滑らかにすることにより、ポルタメント (ある音から別の音に、滑らかにピッチを変化させる奏法) を実現できます。

Waveshaper (波形シェイパー)

入力信号の波形を変形するもので、特性曲線は 2 点で区切られた折れ線になります。

Wrapper (ラッパー)

入力信号を増幅した結果、ある閾値を超えてしまった場合に、超えた部分を閾値の下に折り返すように働きます。オシレーターの出力信号波形を歪ませるのに使うと、Sync あるいはパルス幅変調音のような効果が得られます。

23.11. イベント処理系マクロ

0-1 to 0-127 Range Converter (0 ～ 1 から 0 ～ 127 への範囲変換器)

入力信号に 127 をかけあわせて出力します。「Classic Modular」ではピッチも 0 ～ 1 の範囲で制御しますが、一般には 0 ～ 127 の範囲で変化すると想定されています。そのようなモジュールやマクロと接続する場合は、このマクロで変換してください。また、変調信号を位置イベントに変換して、ステップ・シーケンサーの位置を指示する信号として使うこともできます。

乗算の後、整数値に四捨五入して出力します。

Quantizer (クオンタイザー)

振幅をステップ・サイズの整数倍に量子化 (クオンタイズ) することにより、階段波の形に歪ませます。

Randomizer (ランダマイザー)

入力イベントの値に、所定の範囲の乱数を加えて出力します。

モジュール・リファレンス

モジュール (Module) は、REAKTOR のアンサンブル (Ensemble) を構成する最も基本的な部品です。この章は主として、独自のアンサンブルを一から構築しようという方を対象としています。とはいえ、そのようなことには関心がない、あるいは未経験なので自信がないという方でも心配はいりません。REAKTOR は次のようなさまざまな楽しみ方ができます。

- ・ 独自アンサンブルの構築に関心がない場合は、既存のアンサンブルを使ってみるとよいでしょう。
- ・ 好みのインストゥルメントを組み合わせさせてアンサンブルを構築したい場合は、既存のインストゥルメントを使ってみてください。
- ・ 既存の部品を使って独自のインストゥルメントを作りたい場合は、マクロを組み合わせてください。
- ・ アンサンブルの個々のパラメーターを細かく制御したい場合は、モジュール単位で操作してください。

この章では REAKTOR のモジュールすべてについて、機能の概要を解説するほか、必要に応じて **Propaties** ウィンドウの設定項目や入出力ポートの仕様を説明します。

REAKTOR のオブジェクト、すなわちモジュール、インストゥルメント、マクロ、アンサンブルはいずれも、**Properties** ウィンドウを開くと **LABEL** 欄があります。ストラクチャー (オブジェクトの構成) を表示すると、各オブジェクトを表すアイコン上にこのラベルが表示されます。モジュールのラベルは、初期状態ではその機能を表す文字列になっています。変更も可能ですが、それによってオブジェクトの構成が分かりにくくなってしまわないよう注意してください。

混成モジュール

REAKTOR には**混成**モジュールが多数あります。混成モジュールは、通常はイベント処理モジュールとして扱われ、ポート・ラベルが赤の表示になります。入力ポート上に緑の点があれば、イベント・ケーブルだけではなくオーディオ・ケーブルも接続できることを表します。実際にオーディオ・ケーブルを接続すると、モジュールはオーディオ処理用に変わり、ポート上の点が黒くなるほか、ラベルからもそのことがわかる

ようになっています。一方、イベント・ケーブルを接続した場合は赤い点になります。混成モジュールは次のように動作します。

- ・ 入力ポートにオーディオ・ケーブルとイベント・ケーブルの両方を接続した場合、またはオーディオ・ケーブルのみを接続した場合、モジュールは常にオーディオ・レートで動作します。
- ・ 入力ポートにイベント・ケーブルのみを接続した場合、モジュールはイベント・レートで動作します。
- ・ あるモジュールの出力を他のモジュールのイベント入力につないだ場合、それ自身は自動的にイベント処理モジュールになります。したがって、入力ポートにオーディオ・ケーブルをつなぐことはできません。
- ・ あるモジュールの出力を他のモジュールのオーディオ入力につないだ場合は、入力ポートに接続するケーブルに応じて、オーディオ処理モジュールとしてもイベント処理モジュールとしても動作させることができます。
- ・ 入力側にオーディオ・ケーブルを接続することにより、モジュールがオーディオ・レートで動作するようになった場合、その出力を他のモジュールのイベント入力に接続することはできません。

動的ポート管理

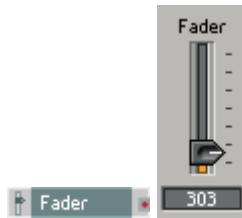
モジュールの機能上、入出力ポート数を可変にすることに意味がある場合は、入出力ポートの**動的**管理機能が有効になります。このようなモジュールに入力ポートを追加したい場合は、**Ctrl** キーを押しながら、追加したいケーブルをマウスでつかんで動かし、既存のポートの上で放してください。新しいポートが追加される位置は、モジュール上にマウス・ポインターを持っていくと表示されます。マウス・ポインター位置 (Y 座標) によって追加される位置が決まるようになっているので、既にあるポートとポートの間に追加することも可能です。

パネル系モジュール

パネル (Panel) 系モジュールには、REAKTOR のさまざまな処理を画面上で制御できるようにする働きがあります。**A** および **B** の制御パネル上に、それぞれ独立に配置することができます。もっぱら画面表示のみを目的とするモジュールもあります。**Lamp**、**Meter**、**Scope** など、REAKTOR の処理状況を表示するもののほか、単なる装飾用に画像や文字列を表示するモジュールがこれにあたります。これ以外のモジュールは一般に、データを生成したり他のモジュールに転送したりして、REAKTOR の動作を制御します。**Fader**、**Knob**、**Button**、**Switch**、**List**、**XY** などがこれにあたり、データを表示すると同時に制御もできるようになっています。

Fader (フェーダー)

パネル系



スライダーを操作して出力値 (イベント / オーディオ) を調整します。値は常にひと通りに決まるので、多声入力につないだ場合、どの声部にも同じ値が渡されることになります。

プロパティ - Function ページ

- ・ **Range**: スライダー操作により調整される値の最小値と最大値を、**Min** および **Max** として設定します。値が変化する刻み幅を **Stepsize** として指定できます。別法として、最小値から最大値の幅をいくつに区切るか、**Num Steps** で指定することも可能です。**Stepsize** と **Num Steps** は、一方を変更すればもう一方も連動して変わるようになっています。刻み数の最大は 127 で、これは **Stepsize** を 0、または **Num Steps** を 127 にした場合に当たります。

注意: **Num Steps** 欄には、MIDI 信号の標準的な分解能である 128 よりも大きな値を設定できます。REAKTOR 内部ではこの高い分解能で処理できるのですが、MIDI データの形で外部とやり取りする段になると分解能が落ちてしまうので注意してください。状況にもよりますが、制作環境が変わると機能や音質に影響が及ぶ可能性があります。一般には、音声処理には MIDI の分解能で充分であり、特に問題は起こりません。しかし、高レゾナンスのフィルターを使っている、カットオフ周波数を変化させたい場合など、内部的には分解能を高くして処理しなければならないこともあるでしょう。

- ・ **Mouse Res:** スライダーを最小値から最大値まで変化させるために、マウス・ポインターを何ピクセル動かさなければならないか、を表します。**Appearance** ページで指定した **Pixel in Y** の 2 倍の値を指定すれば、フェーダーの高さの 2 倍に相当する距離を動かすとスライダーが最小値から最大値まで変化することになります。

Mouse Res よりも **Num Steps** の値の方が大きければ、取りうるすべての値をマウス操作で設定することはできなくなります。逆に **Mouse Res** の方が大きければ、値を 1 単位増やすために 1 ピクセル以上動かす必要があります。

- ・ **Default:** フェーダーを初期化したときの値を表します。具体的には次のような場合にこの値が使われます。
 - ・ **Snapshot** ウィンドウで **Default** ボタンを押すことにより、アンサンブル / インストゥルメントの制御つまみ類をすべてリセットした場合。
 - ・ 2 つの **Snapshot** ウィンドウ間のモーフィング (補間) で、一方のスナップショットを「default」とした場合。制御つまみ類の値は、もう一方のスナップショット上の設定値と、**Default** 欄に設定した値との間で連続的に変化します。
 - ・ インストゥルメントに新たに制御つまみ類を追加した場合、それ以前に保存したスナップショットにはもちろん値が設定されていないので、この **Default** 値を採用します (但し **Snap Isolate** がオンの場合を除く)。当然、この後スナップショットを呼び出してつまみ類を調整すれば値は変わります。
- ・ **Snap Isolate:** オンにすると、スナップショットを呼び出しても値が変わらないようになります。

- ・ **Random Isolate**: オンにすると、スナップショットのランダム化機能を実行しても反応しなくなります。
- ・ **ID for Snapshot Files**: これはスナップショット管理用の ID 番号です。パネル系のモジュールを追加すると、この値が自動的に割り当てられます。番号を変えても構いませんが、保存済みのスナップショットでは元の番号で管理されているので、これを呼び出しても値が反映されません。変更する場合はこの点に注意してください。

プロパティー - Info ページ

- ・ **INFO**: フェーダーの機能を説明する文章を入力します。ヒント機能をオンにすると、マウス・ポインターをフェーダー上に持っていったとき、このテキストがヒントとして表示されるようになります。

プロパティー - Appearance ページ

ラベルを設定して **Visible** をオンにすると、フェーダーに添えてラベルが表示されます。また、**Value** をオンにすると、値も表示されるようになります。さらに、フェーダー自体の画像を非表示にし、値のみ表示されるようにすることもできます。この場合でもマウス操作で値を増減することは可能です。

フェーダーの方向 (縦 / 横) は、**Form** ボタンで選択できます。

フェーダーの大きさは、**Size** 部の **Pixel in Y** 欄で調節できます。

Big、**Medium**、**Small** の 3 種類から幅を選んでも構いません。

プロパティー - Connection ページ

詳しくは 18.5 節「つまみ類に関する **Connection** プロパティー」を参照してください。

Knob (ノブ)

パネル系



外観を除き、上記の **Feder** モジュールと同じ機能です。



ボタンを押すと、これに対応する値 (イベント / オーディオ) が出力されます。オン / オフの時に出力する値を、それぞれ **On Value**、**Off Value** として設定しておくことができます。値は常にひと通りに決まるので、多声入力につないだ場合、どの声部にも同じ値が渡されることになります。

プロパティ - Function ページ

- ・ **Range**: オン / オフの時に出力する値を、それぞれ **On Value**、**Off Value** として設定します。
- ・ **Mode**: 動作モードを次の 3 種類から選択します。
 - ・ **Trigger** モード: ボタンを押したときにのみイベントが生成されます。
 - ・ **Gate** モード: ボタンを放したとき、値 0 のイベントが出力されます。
 - ・ **Toggle** モード: ボタンを押すたびにオン / オフが切り替わります。
- ・ **Default = On**: オンにすると、初期化操作を施したとき、ボタンを押した状態になります。
- ・ **Snap Isolate**: オンにすると、スナップショットを呼び出しても値が変わらないようになります。
- ・ **Random Isolate**: オンにすると、スナップショットのランダム化機能を実行しても反応しなくなります。
- ・ **ID for Snapshot Files**: フェーダーの場合と同様です。

プロパティ - Info ページ

- ・ **INFO**: ボタンの機能を説明する文章を入力します。ヒント機能をオンにすると、マウス・ポインターをボタン上に持っていったとき、このテキストがヒントとして表示されるようになります。

プロパティー - Appearance ページ

Label を設定しておく、ボタンの上にそのラベルが表示されます。
また、ボタンの下に、現在の設定値が表示されます。

ボタンの大きさは、**Small**、**Medium**、**Big** の 3 種類から選べます。

プロパティー - Connection ページ

詳しくは 18.5 節「つまみ類に関する **Connection** プロパティー」を参照してください。

List (リスト / メニュー)

パネル系



画面上にリスト、ドロップ・ダウン・メニュー、選択ボタンを表示するほか、スクロール可能なテキスト表示にも使えます。

プロパティー - Function ページ

- ・ **Append, Insert, Delete:** リストに列挙する項目を追加、挿入、削除します。各項目を選択したときにモジュールから出力される値も、**Value** として指定できます。
- ・ **NUM Entries:** リストに列挙する項目数。

各項目の値が一定の間隔で増えていく (等差数列) よう、ひとまとめに設定することができます。例えば 4 つずつ増えていくようにするのであれば、**Stepsize** 欄に「4」と入力し、**Apply** ボタンを押してください。

- ・ **Mouse Resolution:** マウスの動きの分解能を表します。表示スタイルが **Spin** の場合、マウスでクリックして上下に動かすことにより値を増減できますが、その感度を調整します。

プロパティー - Appearance ページ

リスト / メニューの表示形式を、次の中から選択してください。

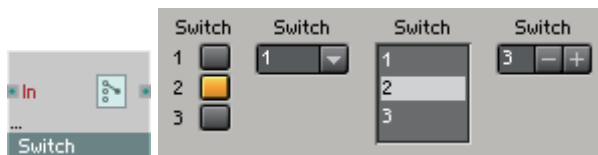
- ・ **Button:** 各項目に対応してボタンが表示されます。インストゥルメント制御パネルでは、ボタンが縦に並びます。オンになっているボタンは、インストゥルメントの **Indicator** として指定された色になります。
- ・ **Menu:** 各項目が列挙されたドロップ・ダウン・リストが表示されます。
- ・ **Text Panel:** 複数行から成るテキストが一度に表示されます。**Size X**、**Size Y** で指定した大きさに入りきらない場合は、スクロール・バーが現れます。
- ・ **Spin:** 各項目についてスピンつまみが表示されます。右側にある **+**、**-** ボタンで値を増減できます。
Size X、**Size Y** 欄に、画面上の大きさを指定してください。

プロパティー - Ports ページ

- ・ **Out:** ある項目を選択したとき、それに対応する値をイベントとして送信するポート番号を指定します。

Switch (スイッチ)

パネル系



信号の流れを切り替えるモジュールです。これ自身で信号を処理することはできず、他のモジュールと接続して信号経路を切り替えるだけの役割です。ボタンまたはリストで入力ポートを1つ選ぶと、その入力ポートから出力ポートに信号を転送するようになります。他の入力ポートでは信号を遮断します。

その結果無効になった入力ポートにつながっているモジュールは、CPUの負荷を軽減するため、自動的に無効になります。該当するモジュールの状態ランプは消灯します。

これは混成モジュールであり、接続(結線)によってイベント信号とオーディオ信号のどちらでも処理できます。また、入力ポートの動的管理機能がついています。

プロパティ - Function ページ

- ・ **Enable Switch Off:** オンにすると、「どの入力ポートもオフ」という状態に切り替えることができます。表示スタイルが **Button** の場合は、オンになっているボタンをもう一度押すとすべてのボタンがオフになります。それ以外の表示スタイルの場合は「Off」という選択肢が追加されます。
- ・ **Min Num Port Groups:** ポート数の最小値をここで指定すると、ケーブル数に関係なく、その個数のポートができます。
- ・ **Mouse Resolution:** マウスの動きの分解能を表します。表示スタイルが **Spin** の場合、マウスでクリックして上下に動かすことにより値を増減できますが、その感度を調整します。

プロパティ - Appearance ページ

Label を設定しておくと、ボタンの上にそのラベルが表示されます。ボタンの大きさは、**Small**、**Medium**、**Big** の 3 種類から選べます。

2つの入力ポートのみを切り替える場合、**1 Toggle Button** をオンにすると、1つめの選択肢に対応するボタンしか表示されなくなります。これがオンならば1番入力、オフならば2番入力が有効になります。

パネル上の表示スタイルは次の中から選択できます。

- ・ **Button:** 各項目に対応してボタンが表示されます。インストゥルメント制御パネルでは、ボタンが縦に並びます。オンになっているボタンは、インストゥルメントの **Indicator** として指定された色になります。
- ・ **Menu:** 各項目が列挙されたドロップ・ダウン・リストが表示されます。
- ・ **Text Panel:** 複数行から成るテキストが一度に表示されます。**Size X**、**Size Y** で指定した大きさに入りきらない場合は、スクロール・バーが現れます。
- ・ **Spin:** 各項目についてスピンつまみが表示されます。右側にある **+**、**-** ボタンで値を増減できます。

Lamp (ランプ)

パネル系



単声信号の値 (真数目盛) によって点灯 / 消灯するインジケーター・ランプ。

入力信号の値が所定の範囲 (**Min** 超かつ **Max** 以下、真数で指定) 内にあるとき、ランプが点灯します。25Hz の信号に同期し、その周期ごとに入力信号の値を調べるようになっています。

Continuous モードをオンにすると、ランプの表示色が値に応じて徐々に変化するようになります。

ランプの色は、**Properties** ウィンドウで、赤、緑、青、黄、インジケーターの中から選択できます。**Indicator Color** をオンにした場合は、インストールメント側でインジケーター色として指定した色になります。

Set On Color ボタン、**Set Off Color** ボタンで、On/Off 時の色を独自に指定することも可能です。

Has Frame をオフにすると、ランプの位置をパネルの境界に隙間なく正確に合わせることができます。

Label を設定しておくと、ボタンの上にそのラベルが表示されます。

Level Lamp (レベル・ランプ)

パネル系



単声信号の値 (対数目盛) によって点灯 / 消灯するインジケーター・ランプ。

値の範囲を対数目盛 (dB 単位) で表すことを除き、**Lamp** と同じです。

RGB Lamp (RGB ランプ)

パネル系



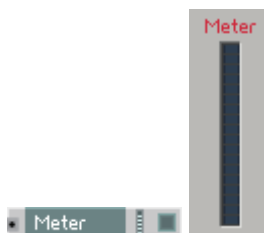
色の変化により状態を表す、大きさ可変のランプです。入力ポートが3つあり、その入力信号によって色が決まります。縦横の大きさは **Appearance** ページで、ピクセル単位で指定できます。

- ・ **R**: 3 原色のうち赤の比率を表すオーディオ入力です。範囲は 0 ~ 1 とします。

- ・ **G**: 3 原色のうち緑の比率を表すオーディオ入力です。範囲は 0 ～ 1 とします。
- ・ **B**: 3 原色のうち青の比率を表すオーディオ入力です。範囲は 0 ～ 1 とします。

Meter (メーター)

パネル系



信号値 (真数目盛) を表示するメーター。

入力信号の値 (真数目盛) を表示します。25Hz の信号に同期し、その周期ごとに入力信号の値を調べるようになっています。

メーターの色は、**Properties** ウィンドウで、赤、緑、青、黄、インジケータの中から選択できます。**Indicator Color** をオンにした場合は、インストールメント側でインジケータ色として指定した色になります。

Set On Color ボタン、**Set Off Color** ボタンで、メーターの上下の部分の色をそれぞれ独自に指定することも可能です。

Number Of Segments は、メーターの最大値と最小値の間をいくつに分割するか (分解能) を指定します。

Size X (Segment)、**Size Y (Segment)** は、メーターの最大値と最小値の間を分割した 1 区間の大きさを表します。メーター全体の高さは、**Size Y (Segment)** と **Number of Segments** の積として求められることになります。

Label を設定しておくと、ボタンの上にそのラベルが表示されます。

LevelMeter (レベル・メーター)

パネル系



信号値 (対数目盛) を表示するレベル・メーター。
値を対数目盛 (dB 単位) で扱うことを除き、**Meter** と同じです。

Picture (ピクチャー)

パネル系

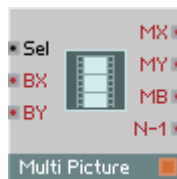


制御パネルの装飾用に、24 ビット BMP 形式または 32 ビット TGA 形式のビットマップ画像を表示します。データの入出力機能はありません。画像の拡大 / 縮小はできませんが、表示する範囲は **Appearance** ページの **Size** 欄で変更できます。

Save bitmap with ensemble をオンにすれば、画像自体もアンサンブル・ファイル内に保存されるようになります。

Multi Picture (マルチ・ピクチャー)

パネル系



Multi Picture モジュールは (XY と同じ) 2 次元のコントローラーで、マウス・ポインターの位置とボタンの状態を出力します。複数の画像を並べてアニメーション表示することも可能です。**Picture Properties** ウィンドウで、フレームの数と、画像をフレームに分割する方向 (縦または横) を指定してください。

TGA 形式 (非圧縮) のビットマップはアルファ・チャンネルにも対応しているので、画像のある部分をマスクし、それ以外の部分を透明にす

ることができます。例えば矩形の背景上に丸いノブを描画したい場合などに有用でしょう。

ピクチャー・ファイルは、**Properties** ウィンドウの **Select-Picture** ドロップ・ダウン・メニューで選択します。ファイルを開くと自動的に **Picture Properties** ウィンドウが開き、ここで必要な設定ができるようになっています。同じピクチャー・ファイルを複数のモジュールで共有することも可能です。

- ・ **Sel**: 表示するフレーム番号を指定するオーディオ入力。最小値は 0、最大値は末尾フレームの番号です。
- ・ **MX**: マウス・ポインターが画像上にある場合の、X 軸方向の位置を表すイベント出力。
- ・ **MY**: マウス・ポインターが画像上にある場合の、Y 軸方向の位置を表すイベント出力。
- ・ **MB**: マウス・ボタンの状態を表すイベント出力 (放した状態ならば 0、押した状態ならば 1)。
- ・ **N-1**: **Picture Properties** ウィンドウで指定したフレーム数から 1 を引いた値を出力。

Text (テキスト)

パネル系

A small rectangular icon with the word "Text" in white text on a dark background.

信号の処理は行わず、もっぱらストラクチャー上に文字列を表示するために使います。例えばインストゥルメントの製作者や日付、使い方などを書いておくとよいでしょう。

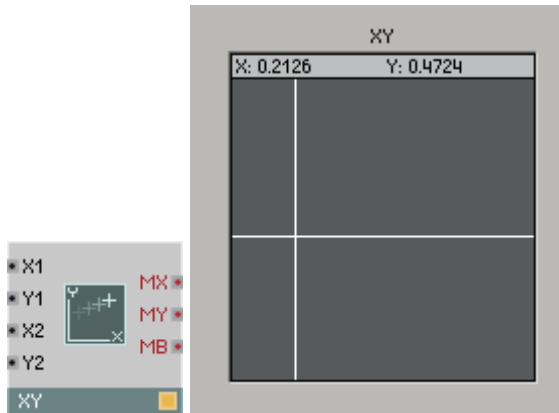
Multi Text (マルチ・テキスト)

パネル系

A small rectangular icon with the text "Multi Text" in white, followed by a small square icon with a plus sign and another small square icon with a minus sign.

何通りかのテキストを用意しておき、切り替えながら表示するためのモジュールです。**Properties** ウィンドウでいくつでもテキストを追加できます。編集や削除にもこのウィンドウを使います。

- ・ **In**: 何番のテキストを表示するか、を指定するオーディオ入力。



オーディオ入力信号を2次元グラフの形で表示するとともに、マウスで操作する2次元コントローラーとしても機能します。

プロパティ - Function ページ

- ・ **Always Active:** オンにすると、どの入力ポートからの信号で制御するかを選ぶ、信号分岐機能が使えるようになります。
- ・ **Incremental Mouse Mode:** オンにすると、ノブと同じように動作するようになります。すなわち、コントローラー上のある位置をクリックしてからマウスを動かした場合、マウス・ポインターの位置から値が直接決まる (絶対値) のではなく、元の値に加算して決める (相対値) 形になります。

プロパティ - Appearance ページ

オーディオ信号の表示方式を、**Pixel**、**Cross**、**Bar**、**Rectangle** から選択します。**Pixel**、**Cross** の場合は、点や十字の表示位置が、**X1** と **Y1** に与えたオーディオ信号の値によって決まります。**Bar**、**Rectangle** ならば、棒や矩形のもう一方の隅の座標が、**X2** と **Y2** に与えたオーディオ信号の値によって決まります。

オーディオ信号の変化が速い場合は **Scope** モジュールを使ってください。**XY** はゆっくりと変化する信号の表示を想定したモジュールです。

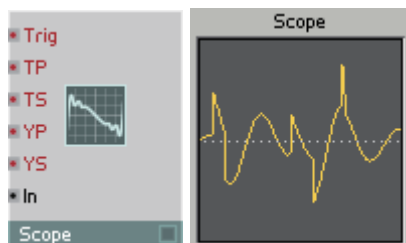
また、マウス位置を表す十字ポインターの大きさも指定できます。

描画された点や十字、矩形などの信号図形は、ゆっくりと消えていきます。消えるまでの時間を **Fade Time** で設定できます (最大値 99)。

- ・ **X1**: 信号図形の X1 座標を決めるオーディオ信号を入力。
- ・ **Y1**: 信号図形の Y1 座標を決めるオーディオ信号を入力。
- ・ **X2**: 信号図形の X2 座標を決めるオーディオ信号を入力。
- ・ **Y2**: 信号図形の Y2 座標を決めるオーディオ信号を入力。
- ・ **MX**: マウス・ボタンを押したとき、そのカーソル位置 (X 座標) を表すイベント信号を出力。
- ・ **MY**: マウス・ボタンを押したとき、そのカーソル位置 (Y 座標) を表すイベント信号を出力。
- ・ **MB**: マウスの左ボタンの状態を出力 (1: 押した状態、0: 放した状態)。

Scope (スコープ)

パネル系



オーディオ信号の波形を時間軸に沿って表示するオシロスコープ。

In に与えられたオーディオ信号の波形を、トリガー入力 (**Trg**) に与えられたイベントに同期して掃引、表示します。波形データをバッファに保持しているので、トリガーがなくてもそのまま表示可能です。時間軸方向、振幅方向の目盛は、制御信号で調整できます。

表示領域の大きさは **Pixel in X**、**Pixel in Y** プロパティ (**Appearance** ページ) で設定します。

データ保持用に確保するバッファ量を、**Function** ページで設定できます (ミリ秒単位)。

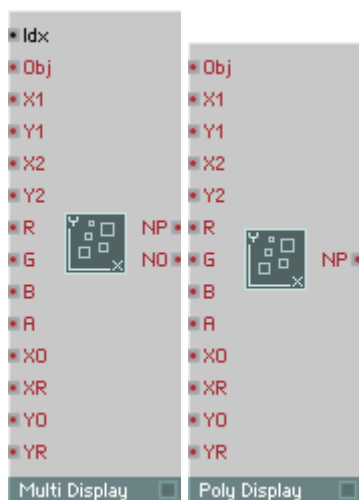
通常は、**A to E Trig** モジュールを **Trg** 入力に接続してトリガー信号とすればよいでしょう。

- ・ **Trg**: トリガー信号を与える単声イベント入力。これに同期して波形を表示します。

- ・ **TP**: 表示する波形を時間軸方向にずらすための、オフセット値 (ミリ秒単位) を表す単声イベント入力。トリガー信号から **TP** ミリ秒後の時点が、表示範囲の左端時刻に当たります。
- ・ **TS**: 時間軸方向の表示幅を制御するための単声イベント入力。表示範囲の左端から右端までの幅で、**TS** ミリ秒分を表します。
- ・ **YP**: 表示する波形を振幅方向にずらすための、オフセット値を表す単声イベント入力。表示範囲の下端が **YP**=-1、上端が **YP**=+1 に当たります。
- ・ **YS**: 振幅方向の表示幅を制御するための単声イベント入力。表示範囲の上端の値と下端の値の差が $2 \times \text{YS}$ に当たります。**YP**=0 であれば、**-YS** ~ **+YS** の範囲が表示されることになります。
- ・ **In**: 表示する信号を与える単声オーディオ入力。

Multi Display、Poly Display

パネル系



いずれも、複数の画像オブジェクト (マーカー、矩形、静止画、動画など) を、切り替えながら表示するためのモジュールです。位置や大きさ、色などのパラメーターは、画像オブジェクトごとに個別に設定できます。

2つのモジュールでは、画像オブジェクトの個数の指定方法が異なります。**Multi Display** ではプロパティの **Number of Objects** で指定しますが、**Poly Display** の場合は声部数がそのまま画像オブジェクトの個数を表します。

したがって **Multi Display** は、表示する画像オブジェクトの個数を、声部数とは独立に設定できることになります。各オブジェクトの識別には **Idx** 入力を使います。これに対して **Poly Display** の場合は、各声部の信号によってオブジェクトを切り替えるため、インデックスの計算が不要なのでプログラムが簡潔になります。その代わり、声部数を超える画像オブジェクトは扱えません。

背景色、背景画像など、表示方法に関するさまざまなプロパティを備えています。

Mouse Area、**Snap Value Array** モジュールと組み合わせて使えば、非常に洗練された、使いやすいユーザー・インターフェイスを実現できます。

Multi Display モジュールの入力はすべて単声イベント信号です。**Idx** には単声オーディオ信号を与えることもできます。

- ・ **Idx**: 画像オブジェクトを識別するインデックスで、(0 ではなく)1 から始まる番号です。小数が与えられた場合は四捨五入して整数として扱います。範囲外の値が与えられた場合の動作は、**Index Behaviour** プロパティの設定によります。複数の画像オブジェクトを重ねて表示する場合は、インデックスの大きい方が手前、小さい方が奥になります。信号を与える順序は重要で、**Idx** をまず指定し、そのあと他のポートにイベントを与える必要があります。
- ・ **Obj**: 画像オブジェクトの種類。
 - ・ -4: 十字マーカー。
 - ・ -3: 直線。起点を **X1**、**Y1** で指定。終点は次の直線オブジェクトの **X1**、**Y1**。
 - ・ -2: 直線。起点を **X1**、**Y1**、終点を **X2**、**Y2** で指定。
 - ・ -1: 棒。
 - ・ 0: 矩形。
 - ・ 1 ~ NP: 静止画のインデックス、または動画のコマ番号 (NP は総コマ数)。

小数が与えられた場合は四捨五入して扱います。

Ignore Index Obj プロパティがオンであれば、画像オブジェクトの種類はすべて同じになります。

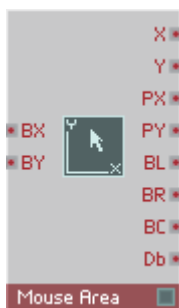
- ・ **X1**、**Y1**: 画像オブジェクトの位置。**Center to X1, Y1** プロパティがオフならば左上隅、オンならば中央を表します。
- ・ **X2**、**Y2**: 画像オブジェクトの右下隅座標。

- ・ **R、G、B**: 画像オブジェクトの色。赤、緑、青のそれぞれについて、混ぜ合わせる割合を 0～1 の範囲で指定します。**Ignore Index RGB** プロパティーがオンであれば、画像オブジェクトの色はすべて同じになります。
- ・ **A**: 画像オブジェクトの透明度 (0= 完全に透過、1= 不透過)。
- ・ **XO、YO**: 表示領域の左下隅の座標。これを調整することによりスクロールを実現します。初期状態を **X Origin、Y Origin** プロパティーで指定しておき、実行時にはこの入力信号で調整します。
- ・ **XR、YR**: 表示領域の右端座標と左端座標の差、上端座標と下端座標の差。これを調整することにより拡大 / 縮小を実現します。初期状態を **X Range、Y Range** プロパティーで指定しておき、実行時にはこの入力信号で調整します。
- ・ **NP**: 動画で現在表示しているコマ番号を出力。
- ・ **NO**: 画像オブジェクトの個数を出力。

Poly Display モジュールの入出力も **Multi Display** とほとんど同じですが、**Idx** 入力、**NO** 出力はありません。また、**XO、XR、YO、YR** を除き、多声イベント信号を与えることができます。

Mouse Area (マウス領域)

パネル系



マウス操作 (クリック、ドラッグ、位置移動) を検出するためのモジュールです。外枠や色を設定できるので、パネル・インターフェイス要素としても使えなくはありません。しかし通常は、透明にして **Multi Display、Poly Display** など他のモジュールの手前に置き、独自のユーザー・インターフェイスを実現します。

X、Y 出力は、絶対座標で出力するか、直前の位置を基準とする相対座標で出力するかを、プロパティーで設定できます。相対座標の場合、

同じ方向へのドラッグ操作を繰り返すごとに順次値が増えていくので、REAKTOR 組み込みのノブやフェーダーと同じように使えます。**BX**、**BY** 入力相対座標の場合のみ意味があり、基準となる座標を設定（初期化）するために使います。通常は **Snap Value** モジュールに接続して、スナップショットに保存しておいた元の基準座標に戻す、という使い方をします。

Outline Style プロパティでモジュールの表示形状を指定できます。**Rectangle** であれば太さ 1 ピクセルの線で矩形の枠を表示、**Bar** であれば領域全体を所定の色で塗りつぶします。

Active State プロパティは、どのように操作すればオフからオンに切り替わるか、を表します。**Selection** ならば **Mouse Area** モジュール領域を選択したとき、**Left/Right/Center** ならば該当するマウス・ボタンを押したときにオンになります。また、**Outline Color** プロパティで、オン / オフ時の色と透明度をそれぞれ設定できます。

さらに、インストゥルメント制御パネル上の位置を、**X offset**、**Y offset** プロパティで設定できます。4 × 4 ピクセル単位の格子を基準として、ずらす幅を表します。

- ・ **BX**: 相対座標の場合に、基準となる X 座標を設定します。**Range X** プロパティで設定した範囲の値を与えてください。
- ・ **BY**: 相対座標の場合に、基準となる Y 座標を設定します。**Range Y** プロパティで設定した範囲の値を与えてください。
- ・ **X**: マウス位置の X 座標を、**Range X** プロパティによって変換した値を出力します。但しこれは、マウス・カーソルが、**Size X**、**Size Y** プロパティで定義される領域内にある場合に限りです。
- ・ **Y**: マウス位置の Y 座標を、**Range Y** プロパティによって変換した値を出力します。但しこれは、マウス・カーソルが、**Size X**、**Size Y** プロパティで定義される領域内にある場合に限りです。
- ・ **PX**: マウス位置の X 座標を、モジュール領域の左端を基準としたピクセル単位で出力します。基準位置よりも左側ならば負の値になります。**X** と違い、**Size X**、**Size Y** プロパティで定義される領域内でも値を出力します。
- ・ **PY**: マウス位置の Y 座標を、モジュール領域の下端を基準としたピクセル単位で出力します。基準位置よりも下側ならば負の値になります。**Y** と違い、**Size X**、**Size Y** プロパティで定義される領域内でも値を出力します。

- ・ **BL**: 左マウス・ボタンの状態。押した状態ならば 1、放した状態ならば 0。
- ・ **BR**: 右マウス・ボタンの状態。押した状態ならば 1、放した状態ならば 0。
- ・ **BC**: 中マウス・ボタンの状態。押した状態ならば 1、放した状態ならば 0。
- ・ **Db**: ダブル・クリックを検出すると値が 1 のイベントを出力します。その後も値は 1 のままなので、値ではなくイベントの有無を調べて処理するようにしなければなりません。

Stacked Macro (マクロの切り替え表示枠)

パネル系



インストールメント制御パネル内のある領域を、複数の画像オブジェクトで共有するために使います。この領域に表示するオブジェクトは、**Panel Index** モジュールで切り替えます。

ストラクチャーにこのモジュールを挿入した後、パネル上に配置し、プロパティーとして大きさを設定してください。次に、**Stacked Macro** 内に、他のマクロ (複数) と **Panel Index** モジュールを追加します (同じ **Stacked Macro** を追加して入れ子にすることはできません)。この複数のマクロを、**Panel Index** モジュールの入力値に応じて、切り替えながら表示することになります。各マクロのインデックス番号は、インストールメント制御パネル上でそのマクロ上を右クリックしたとき、コンテキスト・メニューに表示されます。

MIDI 入力系モジュール

MIDI 入力系モジュールには、外付け MIDI 機器からの MIDI メッセージを ROKATOR に取り込む働きがあります。MIDI データの種類に応じて、ノート、ペロシティー、ピッチベンド、モノ/ポリ・アフタータッチ、コントローラー、プログラム・チェンジ、MIDI クロックなど別々のモジュールに分かれています。また、REAKTOR には MIDI ノート・イベントの形でゲート・メッセージを生成する機能があり、そのためのモジュールも用意されています。内部接続や OSC 接続で使える MIDI 入力系モジュールもあります。

Note Pitch (ノート・ピッチ)

MIDI 入力系



MIDI ノート・オン・イベントを検出し、ピッチ (第 1 データ・バイト) 情報を出力する多声イベント源。値の範囲は **Min**、**Max** プロパティで設定できます。分解能は 128 段階です。ノート・オフ・イベントには反応しません。

値の範囲を 0～127 とし、発振器系モジュールの **P** 入力に接続すると、平均律で演奏できます。出力値は半音単位で、中央 C が 60 に当たります。値の範囲を変えるとそれに応じてピッチ幅も変わるので、例えば 4 分音 (半音をさらに 2 分割) を出すことも可能です。

Pitchbend (ピッチベンド)

MIDI 入力系



MIDI ピッチベンド・イベントを検出、出力する単声イベント源。分解能は 16384 段階です。ピッチベンド・コントローラーが中立位置にあるとき、出力値は 0 になります。値の上限と下限は、**Max**、**Min** プロパティで独立に設定できます。

発振器系モジュールを制御する場合は、**Max=1**、**Min=-1** とし、ピッチ (ノート番号) にピッチベンド値を加算して **P** 入力に与えてください。別の使い方として、**Max=12**、**Min=-12** とすると、上下 1 オクターブの範囲で大きくピッチを変化させることができます。

Gate (ゲート)

MIDI 入力系



MIDI ノート・オン / オフ・イベントを検出、出力する多声イベント源。出力値はノート・オン・イベントのベロシティー値で決まります。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階です。ノート・オフ・イベントを検出すると値は 0 になります。**Min** と **Max** を同じ値にすれば、ベロシティー値にかかわらず、ノート・オン時の出力値は一定になります。

Single Trig. Gate (シングル・トリガー型ゲート)

MIDI 入力系



MIDI ノート・オン / オフ・イベントを検出、出力する単声イベント源。シングル・トリガー型である点を除き、**Gate** と同じです。

「シングル・トリガー」とは、あるキーを押したままの状態での別のキーを叩いた場合 (レガート奏法)、後続のキーに対応するイベントは出力されない、という特性です。エンベロープ系モジュールを制御するならば、多くの場合この「シングル・トリガー」特性が必要です。

Sel. Note Gate (ノート番号選択型ゲート)

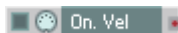
MIDI 入力系



特定のノート番号の MIDI ノート・オン / オフ・イベントを検出、出力する単声イベント源。特定の番号のノート・イベントのみ検出する点を除き、**Gate** と同じです。

On Velocity (オン・ベロシティー)

MIDI 入力系



MIDI ノート・オン・イベントを検出し、ベロシティー (第 2 データ・バイト) 情報を出力する多声イベント源。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階です。

Off Velocity (オフ・ベロシティー)

MIDI 入力系



ノート・オフ・イベントを検出する点を除き、**On Velocity** と同じです。もっとも、キーを放したときに 0 以外のベロシティ値を返す MIDI キーボードはほとんどありません。

Controller (コントローラー)

MIDI 入力系



MIDI コントロール・チェンジ・イベントを検出し、コントロール・チェンジ (第 2 データ・バイト) 情報を出力する単声イベント源。検出するコントロール・チェンジ番号は **Control No** プロパティーで指定します。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階です。

MIDI コントローラーの状態をスナップショットとして保存しておけば、後で呼び出して復元することができます。但し **Snap Isolate** をオンにすると、スナップショットを呼び出しても状態が変わらないようになります。

このモジュールの出力は単声で、イベント信号としてもオーディオ信号としても使えます。

よく使う MIDI コントロール・チェンジ番号としては、次のようなものがあります。

- ・ 1: モジュレーション・ホイール
- ・ 2: ブレス・コントローラー
- ・ 7: ボリューム
- ・ 10: パン
- ・ 64: サスティン・スイッチ
- ・ 65: ポルタメント・スイッチ
- ・ 66: ホールド・スイッチ (ソスティヌート)

個別の MIDI キーボードで出力可能なコントロール・チェンジ番号については、付属の MIDI インプリテーション・チャートを参照してください。

Ch. Aftertouch (チャンネル・アフタータッチ)

MIDI 入力系



MIDI チャンネル・アフタータッチ・イベントを検出し、プレッシャー情報を出力する単声イベント源。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階です。

Poly Aftertouch (ポリ・アフタータッチ)

MIDI 入力系



MIDI ポリ・アフタータッチ・イベントを検出し、プレッシャー情報を出力する単声イベント源。叩いたキーに対応する音を、REAKTOR インストゥルメントの特定の声部で鳴らす場合にのみ有効です。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階です。但しこのイベントに対応した MIDI キーボードはほとんどありません。

Sel. Poly AT (ノート番号選択型ポリ・アフタータッチ)

MIDI 入力系



所定のノート番号の MIDI ポリ・アフタータッチ・イベントを検出し、プレッシャー情報を出力する単声イベント源。ノート番号は **Note No** プロパティー、値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階です。

但しこのイベントに対応した MIDI キーボードはほとんどありません。

Program Change (プログラム・チェンジ)

MIDI 入力系



MIDI プログラム・チェンジ・イベントを検出し、プログラム番号を出力する単声イベント源。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階です。

このモジュールを使う場合、インストゥルメントの **Prog. Change Enable** プロパティーはオフにしておく方がよいでしょう。オンになっていると、MIDI プログラム・チェンジ・イベントによりスナップショットが呼び出されてしまいます。

Start/Stop (開始 / 停止)

MIDI 入力系



外付け MIDI 機器または内蔵マスター・クロックと同期を取るための、開始 / 停止イベントを生成します。

Start/Stop モジュールの出力は単声ゲート信号です。出力値は **Output Value** プロパティーで指定できます。ツールバーの **Start** ボタンを押すか、MIDI Start イベントを受け取ると、その時点で信号の値が変化します。その後、**Stop** ボタンを押すか MIDI Stop イベントを受け取ると値は 0 に戻ります。

このモジュールは通常、シーケンサーやイベント・ディバイダーのリセット入力に接続し、MIDI クロックと同期を取るために使います。

1/96 Clock (96 分音符クロック)

MIDI 入力系



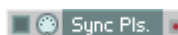
外付け MIDI 機器または内蔵マスター・クロックに反応してクロック信号を発生するモジュールです。

96 分音符に相当する時間ごとにイベントを出力します。出力値は **Output Value** プロパティーで指定できます。

Sync Clock モジュールと違って、出力されるイベントを **Frequency Divider** モジュールで分周することにより、任意の分解能のクロックを生成できます。

Sync Clock (同期クロック)

MIDI 入力系



外付け MIDI 機器または内蔵マスター・クロックの信号を供給するモジュールです。

Sync Clock モジュールの出力は単声ゲート信号です。出力値は **Output Value** プロパティーで指定できます。拍ごとにその値に変化し、ある時間が経過すると 0 に戻ります。開始 / 停止イベントでモジュールを起動 / 停止できます。

クロック信号のレート (4 分音符、8 分音符、16 分音符など) は **Rate** プロパティで指定します。

ゲート信号の長さ (8 分音符相当、16 分音符相当など) は、**Duration** プロパティで指定します。

Song Pos (ソング位置)

MIDI 入力系

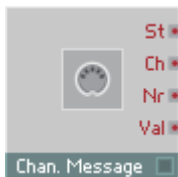


先頭を基準としたソング位置を、96 分音符単位で数えて出力するモジュールです。通常、**Modulo** モジュールの **A** にソング位置、**B** に **Constant** モジュールによる定数 6 を入力して、**Div** 出力から 16 分音符単位で数えたソング位置を取り出します。

- ・ **96**: 96 分音符 (24 ppq. 相当) 単位のソング位置を出力します。ほかの単位に換算したい場合は **Modulo** モジュールを使ってください。
- ・ **96a**: 96 分音符 (24 ppq. 相当) 単位のソング位置を、サンプル単位の精度で出力するオーディオ出力です。

Channel Message (チャンネル・メッセージ)

MIDI 入力系



外付け MIDI 機器 (キーボード、シーケンサーなど) や、アンサンブル内の別のインストゥルメントから、MIDI チャンネル・メッセージを受信、出力する単声イベント源です。情報の出力順序は、出力ポートの並び順 (上から下) に一致します。したがって、イベントの種類 (例: コントロール・チェンジ)、イベント源 (例: チャンネル 1 のコントローラー 7 番) のあとで、実際の値が出力されることになります。

- ・ **St**: 受信したメッセージの種類を出力。
 - ・ 0 = ノート・オフ
 - ・ 1 = ノート・オン
 - ・ 2 = ポリ・アフタータッチ

- ・ 3 = コントロール・チェンジ
- ・ 4 = プログラム・チェンジ
- ・ 5 = チャンネル・アフタータッチ
- ・ 6 = ピッチベンド
- ・ **Ch**: MIDI チャンネル番号 (1 ~ 16)。
- ・ **Nr**: ノート番号、コントロール・チェンジ番号、プログラム・チェンジ番号 (0 ~ 127)。
- ・ **Val**: ノート・イベントの場合はベロシティー値、アフタータッチの場合はプレッシャー、コントロール・チェンジやピッチベンドの場合はその値。外付け MIDI 機器から受信した場合は 7 ビット (ピッチベンドは 14 ビット) にクォンタイズした値ですが、アンサンブル内の別のインストゥルメントから受信した場合は 32 ビット浮動小数点数です。この値を、**Min**、**Max** プロパティーで表される範囲に収まるよう線形変換して出力します。このプロパティーのデフォルト値は 0 ~ 1 ですが、(プログラム・チェンジなど) 状況によっては、相互接続しやすいよう 0 ~ 127 に変更した方がよいでしょう。

MIDI 出力系モジュール

REAKTOR には、外部から受け取った MIDI メッセージを処理するだけでなく、自ら生成する機能もあります。MIDI 出力系モジュールには、メッセージを生成して外付け MIDI 機器に送信する働きがあります。MIDI 入力系モジュールと対応する形で出力系モジュールが用意されているほか、内部接続や OSC 接続で使えるモジュールもあります。

Note Pitch/Gate (ノート・ピッチ / ゲート)

MIDI 出力系



単声または多声のイベント信号を MIDI ノート・イベントに変換、出力するモジュール。ゲート入力 **G** のイベント信号に同期して、REAKTOR の MIDI ポートから MIDI メッセージを出力します。イベント信号の値はベロシティを表し、イベント値 0 ~ 1 がベロシティ値 0 ~ 127 に相当します。イベント値が 0 であれば、ノート・オフ・イベントを出力します。

出力するノート・イベントのピッチは、**P** 入力に指定します。値の範囲は **Min**、**Max** プロパティーで設定できます。最小値が MIDI ノート番号 0、最大値がノート番号 127 に対応します。

Pitchbend (ピッチベンド)

MIDI 出力系



単声イベント信号を MIDI ピッチベンド・イベントに変換、出力するモジュール。REAKTOR の MIDI ポートから MIDI メッセージを出力します。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 16384 段階で、**Min** ~ **Max** の値が -8192 ~ +8191 になるよう線形変換して出力します。

Controller (コントローラー)

MIDI 出力系



単声イベント信号を MIDI コントロール・チェンジ・イベントに変換、出力するモジュール。REAKTOR の MIDI ポートから MIDI メッセージを出力します。出力するコントロール・チェンジ番号は **Control No** プロパティーで指定します。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階で、**Min** ～ **Max** の値が 0 ～ 127 になるよう線形変換して出力します。

Ch. Aftertouch (チャンネル・アフタータッチ)

MIDI 出力系



単声イベント信号を MIDI チャンネル・アフタータッチ・イベントに変換、出力するモジュール。REAKTOR の MIDI ポートから MIDI メッセージを出力します。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階で、**Min** ～ **Max** の値が 0 ～ 127 になるよう線形変換して出力します。

Poly Aftertouch (ポリ・アフタータッチ)

MIDI 出力系



AT 入力値を MIDI ポリ・アフタータッチ・イベント、**P** 入力値をノート番号に変換し、イベントとして出力します。

- ・ **P**: ピッチ入力。 **Min** ～ **Max** の値が 0 ～ 127 になるよう線形変換し、ノート番号として出力します。
- ・ **AT**: アフタータッチ信号入力 (0 ～ 1)。これを 0 ～ 127 の範囲に線形変換し、アフタータッチ値として出力します。

Sel. Poly AT (ノート番号選択型ポリ・アフタータッチ)

MIDI 出力系



単声イベント信号を MIDI アフタータッチ・イベントに変換、出力するモジュール。ノート番号は **Note No** プロパティー、値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階で、**Min** ～ **Max** の値が 0 ～ 127 になるよう線形変換して出力します。

但し MIDI ポリ・アフタータッチ・イベントに対応した MIDI 機器はほとんどありません。

Program Change (プログラム・チェンジ)

MIDI 出力系



単声イベント信号を MIDI プログラム・チェンジ・イベントに変換、出力するモジュール。値の範囲は **Min**、**Max** プロパティーで設定できます。分解能は 128 段階で、**Min** ~ **Max** の値が 0 ~ 127 になるよう線形変換して出力します。

Start/Stop (開始 / 停止)

MIDI 出力系



Start/Continue/Stop のイベントを生成、出力します。

- ・ **G**: ゲート入力。この値が正ならば Start または Continue メッセージ、負または 0 ならば Stop メッセージを出力します。
- ・ **Rst**: リセット入力。イベントとして正の値を与えると、次にゲート入力为正になったときに Start メッセージが出力されます。

1/96 Clock (96 分音符クロック)

MIDI 出力系



96 分音符に相当するクロック・イベントを MIDI 出力から送信します。

- ・ **In**: 正値の信号を与えるとトリガーがかかり、MIDI クロック・イベントの送信が始まります。

Song Pos (ソング位置)

MIDI 出力系



Trig にトリガーがかかった時点で、**Pos** の値を MIDI ソング位置として送信します。

- ・ **Trig**: 正值の信号を与えることによりトリガーがかかります。
- ・ **Pos**: トリガーがかかると、ここに与えた値を (96 分音符単位で計った) ソング位置として送信します。

Channel Message (チャンネル・メッセージ)

MIDI 出力系



外付け MIDI 機器 (シーケンサーなど) や、アンサンブル内の別のインストゥルメントに、MIDI チャンネル・メッセージを送信する単声イベント源です。チャンネル・メッセージは、**St** ポートに与えるイベントに同期して送出されます。したがって、**St** にイベントを与える前に、他の入力に適切な値を与えておく必要があります。なお、どの入力も、単声信号しか受け付けません。

- ・ **St**: メッセージの種類を指定するイベント入力。チャンネル・メッセージはこのイベントに同期して送出されます。
 - ・ 0 = ノート・オフ
 - ・ 1 = ノート・オン
 - ・ 2 = ポリ・アフタータッチ
 - ・ 3 = コントロール・チェンジ
 - ・ 4 = プログラム・チェンジ
 - ・ 5 = チャンネル・アフタータッチ
 - ・ 6 = ピッチベンド
- ・ **Ch**: MIDI チャンネル番号 (1 ~ 16) を設定するオーディオ入力。小数が与えられた場合は四捨五入して扱います。
- ・ **Nr**: ノート番号、コントロール・チェンジ番号、プログラム・チェンジ番号 (0 ~ 127) を表すオーディオ入力。
- ・ **Val**: ノート・イベントの場合はベロシティー値、アフタータッチの場合はプレッシャー、コントロール・チェンジやピッチベンドの場合はその値を表すオーディオ入力。

数学演算系モジュール

数式を決め、REAKTOR に演算させることもできます。演算用モジュールとしては、加減乗除の四則演算のほか、絶対値、逆正接 (ArcTan.)、平方根の逆数など、日常生活ではあまり出てこないものもあります。さらに、指数演算や対数演算により、モジュール入力を目盛を変換することも可能です。例えば、周波数をヘルツ単位の真数目盛で入力するモジュールと、半音単位で測って MIDI ノート番号に変換するなど、対数目盛で管理するモジュールがあるため、両者を変換するモジュールが必要になるのです。

ここで紹介するのはすべて混成モジュールです。すなわち、入力信号に応じ、オーディオ・モジュールとしてもイベント・モジュールとしても使えます。加法、乗法など、入力ポート数を動的に増やせるモジュールも多く、アイコンの左下に小さな点が 3 つついていることで識別できます。**Ctrl** キーを押しながら、入力ポート領域 (モジュール・アイコンの左端) の空いている部分に結線をドラッグすると、新しい入力ポートが自動的に追加されます。

Constant (定数値)

数学演算系



固定値を出力する単声イベント源。値は **Constant Value** プロパティで設定します。イベントは初期設定の際に 1 度送出されるだけです。

Add (加法)

数学演算系



2 つ以上のオーディオ信号またはイベント信号を加算します。出力信号はすべての入力信号の和になります ($\text{Out} = \text{In1} + \text{In2} + \text{In3}$ など)。

チャンネル・レベルがすべて 0dB の、単純なマルチ・チャンネル・ミキサーとしてもつかえます。

Subtract (減法)

数学演算系



2つのオーディオ信号またはイベント信号の差を求めます。1つ目の入力信号から2つ目の入力信号を引いた値が出力されます ($\text{Out} = \text{In1} - \text{In2}$)。

Invert (符号反転、 $-x$)

数学演算系



オーディオ / イベント信号のインバーター。入力信号の符号を反転して出力します。

オーディオ信号は、元の信号と合成するなどしない限り、聴き比べても違いは分かりません。しかし制御信号の反転には大きな意味があります。

Multiply (乗法)

数学演算系



2つ以上のオーディオ / イベント信号の積を求めます。出力信号はすべての入力信号の積になります ($\text{Out} = \text{In1} \times \text{In2} \times \text{In3}$ など)。

一方にオーディオ信号、もう一方に係数を与えることにより、振幅を定数倍するという使い方もできます。すなわち、アナログ・シンセサイザーでいう、電圧制御増幅器 (VCA) に相当する機能です。

入力のいずれか1つ以上に0を与えた場合、出力は常に0になります。

同じ信号を2つの入力に与えれば、平方演算になります ($\text{Out} = \text{In} \times \text{In} = \text{In}^2$)。

また、2つの異なるオーディオ信号を与えて、リング変調の効果を得ることも可能です。

$a * b + c$

数学演算系



乗法と加法の組み合わせで、 $(A * B) + C$ の演算結果を出力します。

Reciprocal (逆数、 $1/x$)

数学演算系



1 を入力値で割った値を出力します。

0 に近い小さな値を入力すると、出力値は非常に大きくなるので注意してください。なお、0 を入力すると出力値も 0 になります。

音声信号に対してこの演算を適用しても、意味のある結果は一般に得られません。0 に近い値を取ることもない、制御信号の処理を想定しています。

Divide (除法)

数学演算系



1 つ目の入力値を 2 つ目の入力値で割った値を出力します。

2 つ目の入力値として 0 に近い小さな値を入力すると、出力値は非常に大きくなるので注意してください。なお、これが 0 の場合は出力値も 0 になります。

除数としてオーディオ信号を与えても、意味のある結果は一般に得られません。

オーディオ信号を被除数として定数やイベント信号の値で割る場合、可能ならば逆数を求めてから掛け算し、同じ結果が得られるようにしてください。CPU に対する負荷を大幅に減らすことができます。

Modulo (剰余)

数学演算系



整数値を 2 つ入力し、その商と余りを求めます。

- ・ **A:** 被除数を与える混成入力。通常の使用範囲: 0 ~ 100。
- ・ **B:** 除数を与える混成入力。通常は整数値を与えますが、そうでなくても構いません。通常の使用範囲: 1 ~ 100。
- ・ **Div:** 商を返す混成出力。A / B を超えない最大の整数です。入力として通常の使用範囲の値を与えた場合の範囲: 0 ~ 100。
- ・ **Mod:** 余りを返す混成出力。剰余を求めることに相当します。範囲: 0 ~ B。

Rectifier (整流、絶対値)

数学演算系



絶対値を求めます。すなわち、入力値が負であれば正になります。

- ・ **In:** 元の信号を与える混成入力。これが負の値であれば、0 からの距離が等しい正の値が出力されます。
- ・ **Out:** 絶対値を返す混成出力。

Rectify/Sign (整流、絶対値 / 符号)

数学演算系



絶対値を求めると同時に、その符号も **Sign** として出力します。

- ・ **In:** 元の信号を与える混成入力。この値の絶対値と符号を出力することになります。
- ・ **Sign:** 入力信号の符号を返す混成出力。正ならば 1、負ならば -1 になります。
- ・ **|x|:** 絶対値を返す混成出力。常に正の値です。

Compare (比較)

数学演算系

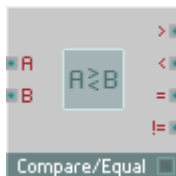


比較関数。2つの入力値を比較し、その結果を出力します。

- ・ **A**: 比較する1つ目の値を与える混成入力。
- ・ **B**: 比較する2つ目の値を与える混成入力。
- ・ **>**: 「AはBよりも大きい」という条件が成り立てば1、成り立たなければ0を返す混成出力。
- ・ **<=**: 「AはBよりも小さいか等しい」という条件が成り立てば1、成り立たなければ0を返す混成出力。

Compare/Equal (比較)

数学演算系

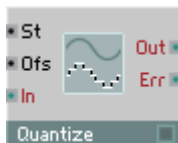


比較関数。2つの入力値を比較し、その結果を出力します。

- ・ **A**: 比較する1つ目の値を与える混成入力。
- ・ **B**: 比較する2つ目の値を与える混成入力。
- ・ **>**: 「AはBよりも大きい」という条件が成り立てば1、成り立たなければ0を返す混成出力。
- ・ **<**: 「AはBよりも小さい」という条件が成り立てば1、成り立たなければ0を返す混成出力。
- ・ **=**: 「AはBと等しい」という条件が成り立てば1、成り立たなければ0を返す混成出力。
- ・ **!=**: 「AはBと等しくない」という条件が成り立てば1、成り立たなければ0を返す混成出力。

Quantize (量子化 / クォンタイズ)

数学演算系



オーディオ信号、イベント信号を、ある離散値に量子化します。
所定の刻み幅の離散値のうち、入力信号の値に最も近いものを出力します。

- ・ **St**: 刻み幅を与える混成入力。0 を与えれば量子化は起こりません。
- ・ **In**: 量子化すべき値を与える混成入力。
- ・ **Out**: 量子化した値を返す混成出力。
- ・ **Err**: 量子化誤差を返す混成出力。 $\text{Err} = \text{Out} - \text{In}$ 。

Expon. (A) (指数演算 - 振幅)

数学演算系



対数目盛で表されたレベル値 (dB) を真数 (振幅) に変換します。

- ・ **Lvl**: 対数目盛 (dB) のレベル値を与える混成入力。通常の使用範囲: -50 ~ 10。
- ・ **A**: 真数値で表された振幅値を返す混成出力。

Expon. (F) (指数演算 - 周波数)

数学演算系



対数目盛で表されたピッチの値 (半音単位の MIDI ノート番号) を真数 (周波数) に変換します。

- ・ **P**: 対数目盛 (半音単位) のピッチの値を与える混成入力。使用範囲: 0 ~ 127。
- ・ **F**: 真数値で表された周波数 (Hz) を返す混成出力。

Log (A) (対数演算 - 振幅)

数学演算系



真数目盛で表された振幅の値を対数 (dB) に変換します。エンベロープの処理などで、時間軸を対数目盛に変換するためにも使えます。

- ・ **A**: 真数目盛の振幅値を与える混成入力。通常の使用範囲: 0 ~ 1000。
- ・ **Lvl**: 対数目盛で表されたレベル値 (dB) を返す混成出力。入力として通常の使用範囲の値を与えた場合の範囲: -60 ~ 0。

Log (F) (対数演算 - 周波数)

数学演算系



真数目盛で表された周波数の値を対数 (半音単位の MIDI ノート番号) に変換します。

- ・ **F**: 真数目盛で表された周波数 (Hz) を与える混成入力。通常の使用範囲: 0 ~ 5000。
- ・ **P**: ピッチの値 (半音単位の MIDI ノート番号) を返す混成出力。範囲: 0 ~ 127。

Power (冪、 x^y)

数学演算系



冪乗。X の Y 乗を計算します (一般に、 X^Y 、 X^Y などと表記)。

- ・ **X**: 基底を与える混成入力。
- ・ **Y**: 指数を与える混成入力。
- ・ **X^Y** : 冪乗を求めて返す混成出力。

Square Root (平方根)

数学演算系

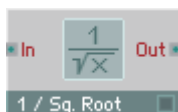


平方根を計算します。

- ・ **In:** 平方根を求めたい値を与える混成入力。なお、負の値を与えた場合、出力は 0 になります。通常の使用範囲: 0 ~ 100。
- ・ **Out:** 平方根を返す混成出力。入力として通常の使用範囲の値を与えた場合の範囲: 0 ~ 10。

1 / Square Root (平方根の逆数)

数学演算系



平方根の逆数を計算します。平方根および逆数の 2 つのモジュールを組み合わせるよりも処理が高速になります。なお、正でない値を与えた場合、出力は 0 になります。

- ・ **In:** 数値を与える混成入力。
- ・ **Out:** 平方根の逆数を返す混成出力。

Sine (正弦)

数学演算系



三角関数の正弦を計算します。定義域、値域とも、-1 ~ 1 の範囲に正規化して扱います。度数単位で表された角度から計算したい場合は、360 で割ってから入力してください。同様に、ラジアン単位の角度から計算したい場合は、 2π (約 6.283) で割ってから入力します。出力は 1 から -1 の範囲で、入力が 0.25 (すなわち 90° 、 $\pi/2$ ラジアン) の場合 1、0.75 の場合 -1 になります。

- ・ **In:** 数値を与える混成入力。
- ・ **Out:** 正弦を返す混成出力。

Sine/Cos (正弦 / 余弦)

数学演算系



三角関数の正弦と余弦を計算します。入力値は、周期 (360 度) が 1.0 であるとして正規化して扱います。

- ・ **In**: 数値を与える混成入力。通常の使用範囲: $-1 \sim 1$ 。
- ・ **Sin**: 正弦を返す混成出力。範囲: $-1 \sim 1$ 。
- ・ **Cos**: 余弦を返す混成出力。範囲: $-1 \sim 1$ 。

Arcsin (逆正弦)

数学演算系



正弦の逆関数です。すなわち $\text{ArcSin}(x)$ は、 -0.25 から 0.25 の範囲で、その正弦が x になるような数を表します。正弦関数の値域が $-1 \sim 1$ に正規化されているのに対応して、逆正弦関数の定義域も $-1 \sim 1$ になっています。入力値が -1 より小さければ -0.25 、 1 より大きければ 0.25 を返します。

- ・ **In**: 数値を与える混成入力。
- ・ **Out**: 逆正弦を返す混成出力。

Arccos (逆余弦)

数学演算系



余弦の逆関数です。すなわち $\text{ArcCos}(x)$ は、 0 から 0.5 の範囲で、その余弦が x になるような数を表します。余弦関数の値域が $-1 \sim 1$ に正規化されているのに対応して、逆余弦関数の定義域も $-1 \sim 1$ になっています。入力値が -1 より小さければ 0.5 、 1 より大きければ 0 を返します。

- ・ **In**: 数値を与える混成入力。
- ・ **Out**: 逆余弦を返す混成出力。



正接の逆関数です (正接は、正弦を余弦で割って得られる関数です)。ArcTan の値域は $-0.25 \sim 0.25$ の範囲で、これは $-90^\circ \sim 90^\circ$ に相当します。

- ・ **In**: 数値を与える混成入力。
- ・ **Out**: 逆正接を返す混成出力。

信号配送系モジュール

信号配送系モジュールには、制御データやオーディオ・データを、REAKTOR のストラクチャーに応じて適切に配送する役割があります。ミキサー、入出力セレクター、遠隔操作スイッチ (リレー)、クロスフェード、パンなどがあります。

Selector/Scanner (セレクター / スキャナー)

信号配送系



Pos 入力の値に応じて信号を選択するモジュールです。**Pos** が整数であれば、対応する番号の入力信号をそのまま出力します。一方、整数でない場合は、その値に近い 2 つの整数値に対応する番号の入力信号を、その小数部分の大きさに応じてクロスフェードするようにミックスして出力します。クロスフェードの処理方法はプロパティーとして設定できます。

Wrap モードを指定すると、**Pos** の値を $(\text{Max} + 1)$ で割った剰余によって信号を選択するようになります。したがって、**Pos** の値として $\text{Max} + 1$ を与えれば 0 を与えたのと同じ、 $\text{Max} + 2$ を与えれば 1 を与えたのと同じことになります。

例えば、**Multitap Delay** の出力信号を入力し、**Ramp Oscillator** の出力を **Pos** に与える、という使い方が考えられます。

入力ポート数は動的に変更できます。**Function** ページの **Min Num Port Groups** プロパティーで指定してください。

- **Pos**: 選択信号を与える混成入力。**Pos** = 0 ならば **In 0**、**Pos** = 1 ならば **In 1** の信号を出力します。**Pos** = 0.5 ならば **In 0** と **In 1** をミックスした信号になります。通常の使用範囲: 0 ~ **Max**。
- **In 0** ~ **In Max**: 被選択信号を与える混成入力。
- **Out**: 選択結果を出力。

Relay (リレー)

信号配送系



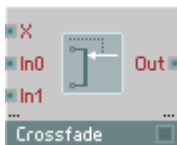
Ctl の値が正ならば上側の入力信号、負または 0 ならば下側の入力信号を出力します。下側の入力信号がない場合は、常に 0 の信号が入力されているとみなします。

In ポートの個数は 1 つまたは 2 つです。**Function** ページの **Min Num Port Groups** プロパティで指定してください。

- ・ **Ctl**: 信号入力を選択するための制御信号。
- ・ **In**: 入力信号。
- ・ **Out**: 選択された信号の出力。

Crossfade (クロスフェード)

信号配送系



In 0 および **In 1** の 2 つの信号をクロスフェードして出力します。混合比率は **X** に与える信号で制御します。

入力ポート数は動的に増やせます。入力ポートの組 (**In 0** および **In 1**) をいくつ設けるか、**Function** ページの **Min Num Port Groups** プロパティで指定してください。

- ・ **X**: 混合比率を制御する混成入力。0 から 1 の範囲で指定してください。0 ならば **In 0**、1 ならば **In 1** の入力信号がそのまま出力されます。
- ・ **In 0**、**In 1**: 入力信号を与える 2 つの混成入力。
- ・ **Out**: ミックス信号の混成出力。 $Out = (1 - X) * In 0 + X * In 1$ 。

Distributor/Panner (信号分配器 / パン)

信号配送系



Pos に与えられた制御信号に従い、入力信号をいくつかの出力に分配します。これによってパンの効果も実現できます。**Pos** が整数であれば、対応する番号の出力ポートにそのまま出力します。一方、整数でない場合は 2 つの出力ポートに振り分けます。その処理方法はプロパティーとして設定できます。

Wrap モードを指定すると、**Pos** の値を $(\text{Max} + 1)$ で割った剰余によって信号を選択するようになります。したがって、**Pos** の値として $\text{Max} + 1$ を与えれば 0 を与えたのと同じ、 $\text{Max} + 2$ を与えれば 1 を与えたのと同じことになります。

出力ポート数は動的に変更できます。**Function** ページの **Min Num Port Groups** プロパティーで指定してください。

- ・ **Pos**: 出力ポートを選択する制御信号入力。
- ・ **In**: 信号入力。
- ・ **Out**: 信号の出力ポート。

Stereo Pan (ステレオ・パン)

信号配送系



音声信号を左右に分配するパン機能のモジュールです。入力信号のレベルを変えて左右に出力することにより、ステレオ効果を実現します。**L** および **R** の出力信号を加算すると、入力信号のちょうど 2 倍のレベルになります。すなわち、1 つの入力信号を、比率を変えて 2 つのポートに出力していることに相当します。

出力ポート数は動的に変更できます。入出力ポートの 3 つ組 (**In**、**L**、**R**) をいくつ設けるか、**Function** ページの **Min Num Port Groups** プロパティーで指定してください。

- ・ **Pan**: パンの制御信号。-1 ならば左、0 ならば中央、1 ならば右になります。
- ・ **In**: 音声信号を与える混成入力。
- ・ **L**: 左チャンネル信号の混成出力。
- ・ **R**: 右チャンネル信号の混成出力。



2つのオーディオ信号のミキサー・モジュール。2つのオーディオ信号を **Lvl** 指定に従ってそれぞれ増幅 / 減衰した後、ミックスして出力します。

入力ポート数は動的に変更できます。入力ポートの組 (**Lvl** および **In**) をいくつ設けるか、**Function** ページの **Min Num Port Groups** プロパティーで指定してください。

- ・ **Lvl**: オーディオ信号の増幅 / 減衰率 (dB 単位)。
- ・ **In**: 増幅するべき入力オーディオ信号。
- ・ **Out**: ミックス信号出力。 $Out = In1 \times 10^{Lvl1/20} + In2 \times 10^{Lvl2/20}$ などとなります。

Stereo Amp/Stereo Mixer (ステレオ・アンプ / ミキサー)



入力信号数が可変のアンプで、パン機能も統合されています。入力信号を **Lvl** 指定に従ってそれぞれ増幅 / 減衰した後、レベルを変えて左右に出力することにより、ステレオ効果を実現します。

入力ポート数は動的に変更できます。入力ポートの3つ組 (**Lvl**、**Pan**、**In**) をいくつ設けるか、**Function** ページの **Min Num Port Groups** プロパティーで指定してください。

- ・ **Lvl**: オーディオ信号の増幅 / 減衰率 (dB 単位)。
- ・ **Pan**: 定位を調整する制御信号。-1.0(左) ~ 1.0(右) の値で指定します。
- ・ **In**: 増幅するべき入力オーディオ信号。
- ・ **L**: 左チャンネルのオーディオ信号出力。
- ・ **R**: 右チャンネルのオーディオ信号出力。

発振器系モジュール

REAKTOR ではさまざまな信号生成器をまとめて発振器 (オシレーター) と呼んでいます。サンプル再生以外の方法による信号生成器は、すべて発振器系モジュールに分類しているのです。正弦波、パルス波、鋸波などの周期波形を生成する狭義の発振器のほか、衝撃波 (インパルス)、階段波 (ステップ波)、雑音 (ノイズ) の発生器、テーブル方式の波形生成器も含まれます。

REAKTOR の発振器系モジュールはすべて、サンプル・レートで決まる上限値まで、任意の周波数の信号を生成できます。したがって、変調信号用の LFO からオーディオ信号生成まで、幅広く使えるのです。あるモジュールを LFO として使い、その出力を変調信号として別のモジュールに与えることも可能ですが、その入力ポートがイベント信号しか受け付けないことがあります (**P** など)。その場合は **A to E (Perm)** モジュールで変換してください。

Sawtooth (鋸波)

発振器系



鋸波発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 鋸波のオーディオ信号出力。

Saw FM (鋸波、周波数変調機能つき)

発振器系

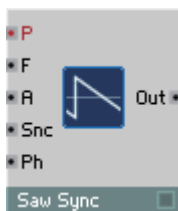


鋸波発振器。周波数は MIDI ノート番号で指定します。振幅変調と周波数変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 鋸波のオーディオ信号出力。

Saw Sync (鋸波、位相同期機能つき)

発振器系



鋸波発振器。周波数は MIDI ノート番号で指定します。位相の同期、振幅変調と周波数変調の機能を備えています。

同期信号入力の立ち上がり (0 から正值への変化) を検出すると、発振信号を **Ph** で指定された位相にリセットします。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで出力信号の位相がリセットされます。
- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ~ 1 の値で表します。0.5 ならば半周期 (π 、 180°)、1 ならば 1 周期 (2π 、 360°) で実質的に 0 と同じことになります。
- ・ **Out**: 鋸波のオーディオ信号出力。

Saw Pulse (鋸波パルス)

発振器系

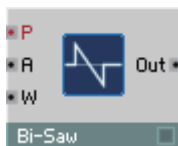


急峻な勾配の鋸波パルス発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。勾配を調整することにより、普通の鋸波からごく短いパルスまで変化させることができます。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Slp**: 勾配を制御するためのオーディオ入力。0 ならば普通の鋸波ですが、値を大きくするほど勾配が急になり、針のようなパルス波を生成できます。
- ・ **Out**: 鋸波のオーディオ信号出力。

Bi-Saw (両極鋸波)

発振器系



両極の鋸波発振器。周波数は MIDI ノート番号で指定します。パルス幅変調、振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: パルス幅変調の変調波信号を与えるオーディオ入力。値は 0 から約 6 の範囲です。0 ならば普通の鋸波ですが、値を大きくするほどパルス幅が短く、振幅が 0 である時間が長くなります。
- ・ **Out**: 鋸波のオーディオ信号出力。

Triangle (三角波)

発振器系

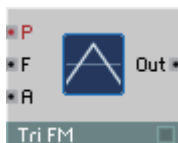


三角波発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 三角波のオーディオ信号出力。

Tri FM (三角波、周波数変調機能つき)

発振器系

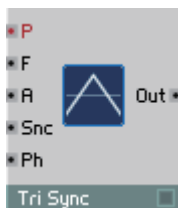


三角波発振器。周波数は MIDI ノート番号で指定します。振幅変調と周波数変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 三角波のオーディオ信号出力。

Tri Sync (三角波、位相同期機能つき)

発振器系



三角波発振器。周波数は MIDI ノート番号で指定します。位相の同期、振幅変調と周波数変調の機能を備えています。

同期信号入力の立ち上がり (0 から正値への変化) を検出すると、発振信号を **Ph** で指定された位相にリセットします。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで出力信号の位相がリセットされます。
- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ~ 1 の値で表します。0.5 ならば半周期 (π 、 180°)、1 ならば 1 周期 (2π 、 360°) で実質的に 0 と同じことになります。
- ・ **Out**: 三角波のオーディオ信号出力。

Tri/Par Symm (三角波 / 放物波、対称性の調整機能つき)

発振器系



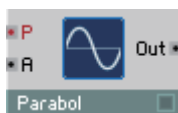
三角波 / 放物波発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。倍音成分をほとんど含まない対称な波形から、広範なスペクトルを含む非対称波形まで調整できます。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。

- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: 対称性を制御するためのイベント入力。-1 ならば上がり勾配の鋸波、0 ならば対称な三角波、1 にすると下がり勾配の鋸波になります。
- ・ **Tri**: 三角波のオーディオ信号出力。
- ・ **Par**: 放物波のオーディオ信号出力。

Parabol (放物波)

発振器系



放物波発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。波形は 2 つの放物線をつなげた形で、正弦波に近いのですが、奇数次の倍音成分がわずかに加わっています。正弦波よりも CPU に対する負荷が小さいので、精度がそれほど必要なければ代用できます。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 放物波のオーディオ信号出力。

Par FM (放物波、周波数変調機能つき)

発振器系



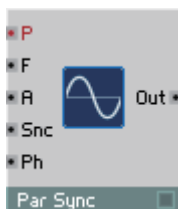
放物波発振器。周波数は MIDI ノート番号で指定します。振幅変調と周波数変調の機能を備えています。波形は 2 つの放物線をつなげた形で、正弦波に近いのですが、奇数次の倍音成分がわずかに加わっています。正弦波よりも CPU に対する負荷が小さいので、精度がそれほど必要なければ代用できます。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。

- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 放物波のオーディオ信号出力。

Par Sync (放物波、位相同期機能つき)

発振器系



放物波発振器。周波数は MIDI ノート番号で指定します。位相の同期、振幅変調と周波数変調の機能を備えています。

同期信号入力の立ち上がり (0 から正值への変化) を検出すると、発振信号を **Ph** で指定された位相にリセットします。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで出力信号の位相がリセットされます。
- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ~ 1 の値で表します。0.5 ならば半周期 (π 、 180°)、1 ならば 1 周期 (2π 、 360°) で実質的に 0 と同じことになります。
- ・ **Out**: 放物波のオーディオ信号出力。

Par PWM (放物波、パルス幅変調機能つき)

発振器系



放物波発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。上半放物線と下半放物線の比率を調整することにより、対称な放物波から、上半のみ、あるいは下半のみの放物線波形まで変化させることができます。

LFO として使う場合に特に有用でしょう。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: 対称性を制御するためのイベント入力。-1 ならば下に開いた放物線波形、0 ならば対称な放物波、1 にすると上に開いた放物線波形になります。
- ・ **Out**: 放物波のオーディオ信号出力。

Sine（正弦波）

発振器系



正弦波発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。

それほど精度が必要ない場合は、CPU に対する負荷が小さい放物波で代用しても構いません。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 正弦波のオーディオ信号出力。

Sine FM（正弦波、周波数変調機能つき）

発振器系



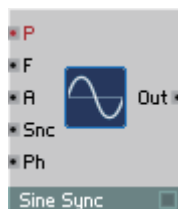
正弦波発振器。周波数は MIDI ノート番号で指定します。振幅変調と周波数変調の機能を備えています。

それほど精度が必要ない場合は、CPU に対する負荷が小さい放物波で代用しても構いません。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 正弦波のオーディオ信号出力。

Sine Sync (正弦波、位相同期機能つき)

発振器系



正弦波発振器。周波数は MIDI ノート番号で指定します。位相の同期、振幅変調と周波数変調の機能を備えています。

同期信号入力の立ち上がり (0 から正値への変化) を検出すると、発振信号を **Ph** で指定された位相にリセットします。

それほど精度が必要ない場合は、CPU に対する負荷が小さい放物波で代用しても構いません。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで出力信号の位相がリセットされます。

- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ～ 1 の値で表します。0.5 ならば半周期 (π 、 180°)、1 ならば 1 周期 (2π 、 360°) で実質的に 0 と同じことになります。
- ・ **Out**: 正弦波のオーディオ信号出力。

Multi-Sine (正弦波の加算合成)

発振器系



4 つの正弦波を加算合成して出力する発振器。正弦波は個別に振幅と周波数を設定できます。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F1/F2/F3/F4**: 1 ～ 4 番正弦波の周波数。**P** で決まる周波数を 1 とした比率で指定します。通常の使用範囲: 0 ～ 20。
- ・ **A1/A2/A3/A4**: 1 ～ 4 番正弦波の振幅 (0 ～ 1)。トレモロやリング変調の効果を出すために使うこともできます。
- ・ **S1/S2/S3/S4**: 加算合成する前の、1 ～ 4 番正弦波の出力。
- ・ **Out**: 4 つの正弦波の合成信号出力。

Pulse (パルス波)

発振器系



パルス波発振器。周波数は MIDI ノート番号で指定します。パルス幅変調、振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: パルス幅を制御するためのオーディオ入力 (-1 ~ 1)。デューティー比は Low:High = $(1+W):(1-W)$ で決まります。したがって、**W**=0 ならば対称な方形波 (50:50)、**W**=-0.33 ならば 33:66、**W**=0.33 ならば 66:33、**W**=0.5 ならば 75:25、**W**=0.8 ならば 90:10 などとなります。
- ・ **Out**: パルス波のオーディオ信号出力。

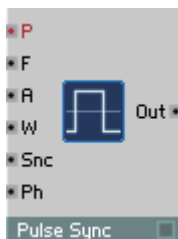
Pulse FM (パルス波、周波数変調機能つき)

発振器系



パルス波発振器。周波数は MIDI ノート番号で指定します。パルス幅変調、振幅変調、周波数変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: パルス幅を制御するためのオーディオ入力 (-1 ~ 1)。デューティー比は Low:High = $(1+W):(1-W)$ で決まります。したがって、**W**=0 ならば対称な方形波 (50:50)、**W**=-0.33 ならば 33:66、**W**=0.33 ならば 66:33、**W**=0.5 ならば 75:25、**W**=0.8 ならば 90:10 などとなります。
- ・ **Out**: パルス波のオーディオ信号出力。



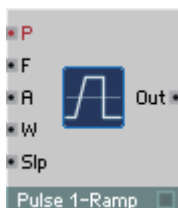
パルス波発振器。周波数はMIDIノート番号で指定します。位相の同期、パルス幅変調、振幅変調と周波数変調の機能を備えています。

同期信号入力の立ち上がり (0 から正値への変化) を検出すると、発振信号を **Ph** で指定された位相にリセットします。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: パルス幅を制御するためのオーディオ入力 (-1 ~ 1)。デューティ比は Low:High = $(1+W):(1-W)$ で決まります。したがって、**W**=0 ならば対称な方形波 (50:50)、**W**=-0.33 ならば 33:66、**W**=0.33 ならば 66:33、**W**=0.5 ならば 75:25、**W**=0.8 ならば 90:10 などとなります。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで出力信号の位相がリセットされます。
- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ~ 1 の値で表します。0.5 ならば半周期 (π 、180°)、1 ならば 1 周期 (2π 、360°) で実質的に 0 と同じことになります。
- ・ **Out**: パルス波のオーディオ信号出力。

Pulse 1-ramp (片台形波)

発振器系

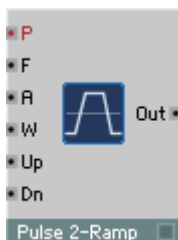


片台形波発振器。パルス波発振器ですが、ある傾斜をもって立ち上がるため、台形の波形（普通のパルス波と鋸波を重ね合わせた波形）になります。周波数はMIDIノート番号で指定します。周波数変調、パルス幅変調、振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: パルス幅を制御するためのオーディオ入力 (-1 ~ 1)。デューティ比は Low:High = $(1+W):(1-W)$ で決まります。したがって、**W**=0 ならば対称な方形波 (50:50)、**W**=-0.33 ならば 33:66、**W**=0.33 ならば 66:33、**W**=0.5 ならば 75:25、**W**=0.8 ならば 90:10 などとなります。
- ・ **Slp**: 立ち上がり部分の勾配を制御するためのオーディオ入力。0 ならば勾配 0、したがって出力は常に 0 のままで、音にはなりません。通常は 1 ~ 20 程度の値で使います。
- ・ **Out**: パルス波のオーディオ信号出力。

Pulse 2-ramp (両台形波)

発振器系



両台形波発振器。パルス波発振器ですが、傾斜をもって立ち上がる / 下がるため、台形の波形になります。周波数は MIDI ノート番号で指定します。周波数変調、パルス幅変調、振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: パルス幅を制御するためのオーディオ入力 (-1 ~ 1)。デューティー比は Low:High = $(1+W):(1-W)$ で決まります。したがって、**W**=0 ならば対称な方形波 (50:50)、**W**=-0.33 ならば 33:66、**W**=0.33 ならば 66:33、**W**=0.5 ならば 75:25、**W**=0.8 ならば 90:10 などとなります。
- ・ **Up**: 立ち上がり部分の勾配を制御するためのオーディオ入力。
- ・ **Dn**: 立ち下がり部分の勾配を制御するためのオーディオ入力。
- ・ **Out**: パルス波のオーディオ信号出力。

Bi-Pulse (両極パルス波)

発振器系



両極パルス波発振器。周波数は MIDI ノート番号で指定します。パルス幅変調、振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **W**: パルス幅を制御するためのオーディオ入力 (0 ~ 1)。**W**=0 ならば方形波、値を大きくするほどパルス幅が狭く、レベル 0 である時間が長くなります。
- ・ **Out**: パルス波のオーディオ信号出力。

Impulse (衝撃波、インパルス)

発振器系



衝撃波発振器。周波数は MIDI ノート番号で指定します。振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 衝撃波のオーディオ信号出力。

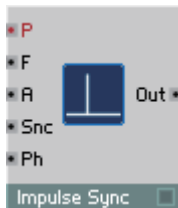
Impulse FM (衝撃波、周波数変調機能つき)

発振器系



衝撃波発振器。周波数は MIDI ノート番号で指定します。周波数変調、振幅変調の機能を備えています。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 衝撃波のオーディオ信号出力。



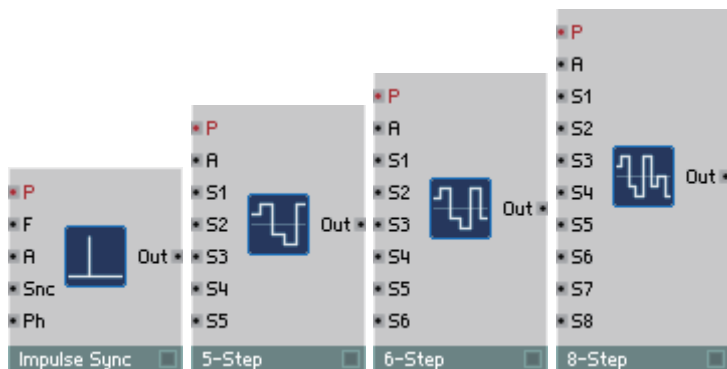
衝撃波発振器。周波数は MIDI ノート 番号で指定します。位相の同期、振幅変調と周波数変調の機能を備えています。

同期信号入力の立ち上がり (0 から正値への変化) を検出すると、発振信号を **Ph** で指定された位相にリセットします。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート 番号で指定し、69 番が 440Hz に相当します。
- ・ **F**: 周波数変調信号を与えるオーディオ入力。Hz 単位の値を与えます。最終的な発振周波数は、**P** で決まる周波数に **F** を加えた値になります。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで出力信号の位相がリセットされます。
- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ~ 1 の値で表します。0.5 ならば半周期 (π 、 180°)、1 ならば 1 周期 (2π 、 360°) で実質的に 0 と同じことになります。
- ・ **Out**: 衝撃波のオーディオ信号出力。

Multi-Step (階段波)

発振器系



4-Step (4段階の階段波)

発振器系

4段階の階段波発振器。周波数は MIDI ノート番号で指定します。各段階のレベルを独立に設定できます。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **S1/S2/S3/S4**: 第 1 ～ 4 段階のレベルを制御するオーディオ入力。
- ・ **Out**: 階段波のオーディオ信号出力。

5-Step (5段階の階段波)

発振器系

5段階の階段波発振器。機能は **4-Step** と同様です。

6-Step (6段階の階段波)

発振器系

6段階の階段波発振器。機能は **4-Step** と同様です。

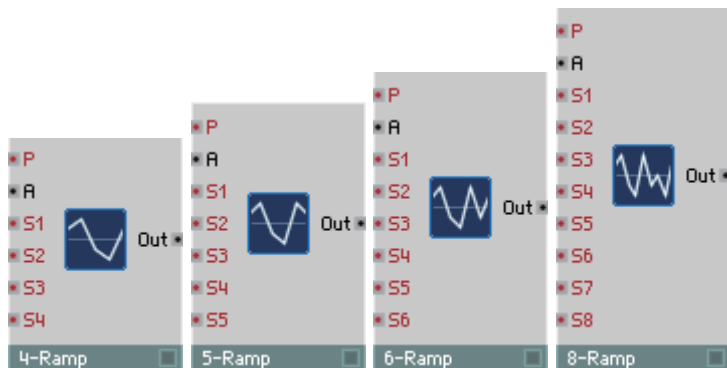
8-Step (8段階の階段波)

発振器系

8段階の階段波発振器。機能は **4-Step** と同様です。

Multi-Ramp (折れ線波)

発振器系



4-Ramp (4 段階の折れ線波)

発振器系

4 段階の折れ線波発振器。周波数は MIDI ノート番号で指定します。各段階のレベルを独立に設定できます。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **S1/S2/S3/S4**: 第 1 ～ 4 段階のレベルを制御するオーディオ入力。
- ・ **Out**: 折れ線波のオーディオ信号出力。

5-Ramp (5 段階の折れ線波)

発振器系

5 段階の折れ線波発振器。機能は **4-Ramp** と同様です。

6-Ramp (6 段階の折れ線波)

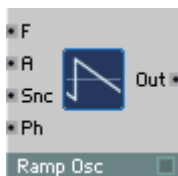
発振器系

6 段階の折れ線波発振器。機能は **4-Ramp** と同様です。

8-Ramp (8 段階の折れ線波)

発振器系

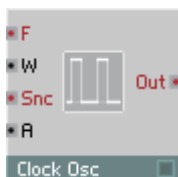
8 段階の折れ線波発振器。機能は **4-Ramp** と同様です。



ランプ波発振器。**Audio Table** モジュールで波形を生成する際に、制御信号として使います。信号レベルが 0 から線形に上昇していき、**A** で指定された値に達すると 0 に戻ります。

- ・ **F**: 発振周波数 (Hz 単位) を指定するオーディオ入力。MIDI ノート番号で指定したい場合は **Expon. (F)** モジュールで変換してください。
- ・ **A**: ランプ信号の振幅を指定します。代表値は 1。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで出力信号の位相がリセットされます。
- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ~ 1 の値で表します。したがってリセット直後のレベルは $A \times Ph$ になります。
- ・ **Out**: ランプ波のオーディオ信号出力。

Clock Oscillator (クロック信号発振器)



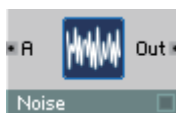
一定周期のクロックを生成します。内蔵の (単声) 発振器により、イベントの形でクロック・オン / オフのパルスが生成されるので、シーケンサー制御用の同期信号などとして利用できます。オン・イベントに対応して出力する値は、プロパティとして設定できます。

- ・ **F**: クロック周波数を Hz 単位で制御するためのイベント入力。テンポを BPM 単位で制御する場合、Hz 単位の値は (拍あたりのクロック数) $\times \text{BPM} / 60$ として求めてください。例えば 1 小節に 4 拍入る拍子で、16 分音符単位、すなわち 1 拍あたり 4 回のクロックを生成したい場合、 $F = 4 / 60 \times \text{BPM} = 0.0667 \times \text{BPM}$ となります。

- ・ **W**: パルス幅を制御するイベント入力 (-1 ~ 1)。クロック・オンの長さでクロック・オフの長さの比率を表し、**W** = 0 ならばオンとオフの長さが等しくなります。オンとオフの比率は Low:High = $(1+W):(1-W)$ で決まります。
- ・ **Src**: 波形の同期信号を与えるイベント入力。これが正になった時点でクロック源と同期します。
- ・ **A**: クロック信号の振幅を制御するための入力。何もつながないと常に 0 値のイベントが生成されてしまい、意味がありません。通常の値: 1。
- ・ **Out**: クロック信号のイベント出力。オン値と 0 の 2 値クロックです。

Noise (雑音発生器)

発振器系

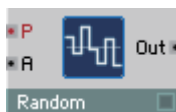


白色雑音の発生器。あらゆる周波数成分が均等な確率で現れます。

- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 雑音のオーディオ信号出力。

Random (乱数信号)

発振器系



乱数信号生成器。**P** で示される周期ごとに、均等な確率で分布する乱数値のレベルに変わります。雑音発生器の後段にサンプル & ホールド回路を置いたものに相当します。

- ・ **P**: 発振周波数を指定するイベント入力。MIDI ノート番号で指定し、69 番が 440Hz に相当します。
- ・ **A**: 振幅を制御するオーディオ入力。出力信号の振幅の最大値を表します。
- ・ **Out**: 乱数のオーディオ信号出力。

Geiger (ガイガー・カウンター)

発振器系

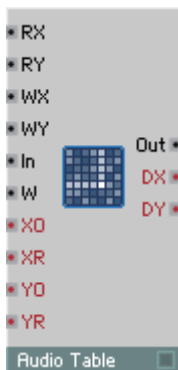


ランダムな間隔でイベント信号を生成し、ガイガー・カウンター (放射能検知器) のような音を作ります。イベントが生成される頻度 (平均値) を **P** で設定します。**Rnd** はランダム化の度合いを表します。

- ・ **P**: 一定時間内に平均何個のイベントを生成するか、その頻度 (時間軸方向の密度) を設定します。頻度値の対数で指定し、通常は -50 ~ 50 程度の範囲で使います。
- ・ **Rnd**: ランダム化の度合いを表します (0 ~ 1)。0 ならば完全にランダム、1 ならば一定周期になります。
- ・ **Clk**: ランダムな時間間隔のクリック音のオーディオ出力。
- ・ **Out**: ランダムな間隔のイベント出力。例えば、エンベロープ系モジュールのトリガー入力 (**G**) に与える、といった使い方があります。

Audio Table (オーディオ・テーブル)

発振器系



一連のデータ値を表の形で保持するモジュール。入力信号をインデックスとみなして表を引き、オーディオ信号として出力します。データ値は、実行時にオーディオ信号の形で与えるほか、事前に画面上に表示し、編集することも可能です。インデックスは、**X** 入力を使って 1 次元で指定する方法と、**X**、**Y** の両方を使って 2 次元で指定する方法があります。

データ値の読み取りは、**RX** および **RY** にインデックスを与えて行います。逆に実行時にデータ値を与えるときは、値を **IN** に与え、**WX** および **WY** にインデックスを指定します。

X は水平位置 (左から右)、**Y** は垂直位置 (上から下) を表し、番号は 0 から始まります。

制御パネルにはデータ値の一部または全部を表示できます。プロパティの設定により、動作をさまざまに調整できます。

プロパティ、メニュー、キーボード・ショートカットについては、第 22 章「テーブル型のモジュール」を参照してください。

- ・ **RX**: データ値の読み取り用インデックス (X 軸方向) を与えるオーディオ入力。
- ・ **RY**: データ値の読み取り用インデックス (Y 軸方向) を与えるオーディオ入力。データ値を 2 次元で扱う場合に使います。
- ・ **WX**: データ値の書き込み用インデックス (X 軸方向) を与えるオーディオ入力。
- ・ **WY**: データ値の書き込み用インデックス (Y 軸方向) を与えるオーディオ入力。
- ・ **W**: データ書き込みのトリガーを与えるオーディオ入力。
- ・ **In**: 書き込むデータ値を与えるオーディオ入力。**W** の値が正になったときに、**WX** と **WY** で示されるセルに書き込まれます。
- ・ **XO**: 制御パネルに表示するインデックス範囲 (X 軸方向) のオフセットを与えるイベント入力。この範囲中のどの位置を表すかは、**View Alignment** プロパティで設定します。また、単位もプロパティで設定できます。
- ・ **XR**: 制御パネルに表示するインデックス範囲 (X 軸方向) の長さ、すなわち表示する個数を与えるイベント入力。
- ・ **YO**: 制御パネルに表示するインデックス範囲 (Y 軸方向) のオフセットを与えるイベント入力。この範囲中のどの位置を表すかは、**View Alignment** プロパティで設定します。また、単位もプロパティで設定できます。
- ・ **YR**: 制御パネルに表示するインデックス範囲 (Y 軸方向) の長さ、すなわち表示する個数を与えるイベント入力。
- ・ **Out**: **RX** と **RY** で示されるセルの値を出力するオーディオ出力。
- ・ **DX**: X 軸方向の表の大きさを出力するイベント出力。
- ・ **DY**: Y 軸方向の表の大きさを出力するイベント出力。

サンプラー系モジュール

オーディオ信号を生成するモジュールのうち、発振器以外のものをサンプラー系に分類します。REAKTOR には、単純なサンプル・プレイヤーに加え、グラニュー方式の音声合成、ピッチや再生速度の増減、拍の分割などの機能を備えた、高度なサンプル・プロセッサが組み込まれています。また、いくつものサンプルを番号で切り替えながら再生するモジュールもあります。

モジュールによっても異なりますが、**Properties** ウィンドウでは次のような設定ができるようになっています。

プロパティ - Function ページ

- ・ **Waveform** ボタン: 波形をかたどったアイコンのボタンで、これを押すとサンプル・マップ・エディターが開きます。
- ・ **Embed Samples In Ensemble**: アンサンプル・ファイルにサンプル・データも保存します。
- ・ **No Stereo**: ステレオ再生を抑止し、左チャンネルのみ使うようにします。これにより CPU の負荷を軽減できます。
- ・ **Quality**: サンプルの再生音質を **Poor**, **Good**, **Excellent** の中から選択するドロップ・ダウン・メニューです。音質を上げれば CPU に対する負荷も増えます。ここでいう音質とは、雑音の有無のことです。もちろん、雑音を生かすことにより、逆に「音楽的」により効果が得られる場合もあるでしょう。
- ・ **Oscil. Mode**: オンにすると発振器モードになります。

発振器モードの場合は、サンプル中に含まれる波形データまたは波形セット (WaveSet) を利用します。波形は 1 周期分の振動を表します。これを繰り返し再生することにより、周期的に振動する状態を作り出すことができるのです。サンプラー・モジュールはこの処理をデジタル的に実現します。発振器系モジュールと同じように、**P** 入力に与えられた信号を MIDI ノート番号として解釈し、そのピッチで周期的に波形全体を再生することによって求める音を作ります。例えば 440Hz と設定した場合、サンプルの長さにかかわらず、全体が毎秒 440 回再生されることになります。

発振器モードが使えるモジュールとしては、**Sampler**、**Sampler FM** があります。

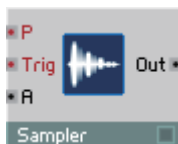
一方 **Sampler Loop** モジュールには波形モードがあります。サンプル全体ではなく、ループ長として表される時間を 1 周期分として扱います。ループ長はサウンド・ファイル中のデータを見て設定しますが、**LL** 入力により変えることもできます。したがってサンプル中に波形がいくつも含まれていても構いません。これを波形セット (WaveSet) と呼びます。例えば 100 周期分の波形が含まれている場合、これを 100 回繰り返して再生すれば、1 万周期分に相当することになります。最初の波形が 0 ~ 99 サイクル、次の波形が 100 ~ 199 サイクル、といった具合です。この場合、いくつもある波形のうちどの部分から再生するかを指定しなければなりません。そこで **LS** 入力によりループの開始位置を制御できるようになっています。波形モードの場合、波形の一部が欠落するようなことのないよう、**LS** の入力値は量子化されます。波形セット中の位置は、ベロシティなどを調整することにより、簡単に制御できます。もちろんこの波形セット・サンプルは、新たに生成するか、またはあるサンプルから派生させる形で用意する必要があります。ライブラリーにはさまざまな波形セットの例が入っています。以上のように、**Sampler Loop** モジュールには、一般的な音声合成機能に加え、波形セットを使った合成方式も統合されています。これと FM 合成とを組み合わせる使えるシステムは、他に例がありません。

プロパティ - Appearance ページ

- **Picture:** このチェックボックスをオンにすると、波形表示が制御パネルに現れます。
- **Size X/Size Y:** 波形を表示する画面上の大きさをピクセル単位で指定します。
- **Scroll Bar:** このチェックボックスをオンにすると、波形表示領域の下にスクロール・バーが現れるようになります。スクロール・バーの中央部分をドラッグすれば表示範囲を動かせるほか、左端または右端の部分を操作すると拡大 / 縮小できます。

Sampler (サンプラー)

サンプラー系



サンプルやサンプル・マップを再生するプレイヤー。多声部に対応し、移調も可能です。

サンプルの管理にはサンプル・マップ・エディターを使います。

Loop プロパティをオンにすると、サンプル全体を繰り返し再生するようになります。トリガー入力に正值のイベントを与えると先頭に戻ります。逆に **Loop** プロパティをオフにすると、トリガー入力に正值のイベントを与えたとき、先頭から末尾まで一度だけ再生するようになります。但し、**Direction** プロパティが **Backward** であれば、末尾から先頭に向かって逆方向に再生します。

Oscillator Mode をオンにすると、**P** 入力で指定された音高になるよう、再生速度を調整するようになります。

- ・ **P**: 再生速度を MIDI ノート番号で指定する制御入力。この指定に応じて再生速度を調整するほか、サンプル・マップの場合はノート番号に合わせてサンプルを切り替えます。サンプルの **Root Key** と同じであれば、そのサンプルの本来の音高で再生されます。
- ・ **Trig**: 正值のイベントを与えると、サンプルの先頭から再生が始まります。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **Out**: サンプルの再生出力。

Sampler FM (サンプラー、周波数変調機能つき)

サンプラー系



サンプルやサンプル・マップを再生するプレイヤー。多声部に対応し、移調も可能です。**F** 入力で再生周波数を変調し、**St** 入力で再生開始位置を調整することができます。

サンプルの管理にはサンプル・マップ・エディターを使います。

Loop プロパティをオンにすると、サンプル全体を繰り返し再生するようになります。トリガー入力に正值のイベントを与えると、**St** で指定される位置に戻ります。逆に **Loop** プロパティをオフにすると、トリガー入力に正值のイベントを与えたとき、先頭から末尾まで

一度だけ再生するようになります。但し、**Direction** プロパティーが **Backward** であれば、末尾から先頭に向かって逆方向に再生します。

Oscillator Mode をオンにすると、**P** 入力で指定された音高になるよう、再生速度を調整するようになります。

- ・ **P**: 再生速度を MIDI ノート番号で指定する制御入力。この指定に応じて再生速度を調整するほか、サンプル・マップの場合はノート番号に合わせてサンプルを切り替えます。サンプルの **Root Key** と同じであれば、そのサンプルの本来の音高で再生されます。
- ・ **F**: 再生速度の変調信号入力。発振器系モジュールの周波数変調と同じ効果です。負の値を与えると再生速度が落ち、値を小さくしていくとついには逆方向に再生されるようになります。
- ・ **St**: トリガー入力にイベントを与えたときの再生開始位置を指定する制御入力。サンプル先頭からのミリ秒単位で指定します。
- ・ **Trig**: 正值のイベントを与えると、サンプルの先頭から再生が始まります。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **Out**: サンプルの再生出力。
- ・ **Lng**: サンプル長 (ミリ秒単位) を表す多声イベント出力。

Sampler Loop (サンプラー・ループ)

サンプラー系



モノラル / ステレオのサンプルやサンプル・マップ、波形セットを再生するプレイヤー。多声部に対応し、移調も可能です。

サンプルの管理にはサンプル・マップ・エディターを使います。

正值のゲート・イベントを与えると、**St** 入力で指定された開始位置からサンプルの再生が始まります。**Loop** プロパティーをオンにすると、再生が始まった後、所定のループ範囲に入るとその範囲を繰り返し再生

するようになります。このループ範囲は、サンプル・ファイル中にあらかじめ設定されていますが、未設定の場合はサンプル全体がループ範囲であるとみなします。但し、**LS**、**LL** 入力で、ループ開始位置とループ長を変更することも可能です。一方、**Loop** プロパティをオフにすると、トリガー入力に正值のイベントを与えたとき、先頭から末尾まで一度だけ再生するようになります。但し、**Direction** プロパティが **Backward** であれば、末尾から先頭に向かって逆方向に再生します。

Loop in Release プロパティをオフにすると、負値のゲート・イベントが与えられたとき、ループ範囲の末尾（逆方向再生ならば先頭）に到達した時点でフェード・アウトするようになります。一方、**Loop in Release** プロパティがオンの場合は、負値のゲート・イベントがあってもほとんど影響がありません。**Loop** プロパティをオフにした場合も同様です。

No Stereo プロパティをオンにすると、ステレオ・サンプルであってもモノラル扱い、すなわち、左チャンネルの信号を **L** と **R** の両方に出力するようになります。CPU に対する負荷が問題になる場合、これをオンにすることにより回避できるかも知れません。

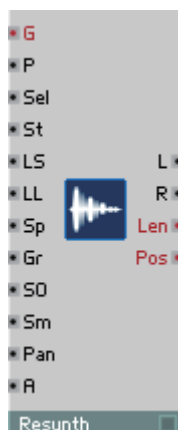
- ・ **G**: この入力に正值のゲート・イベントを与えると、**St** 入力で指定された開始位置から再生が始まります。負値のイベントを与えると、ループ末尾に到達した時点で再生が中断します (**Loop in Release** プロパティがオフの場合)。
- ・ **P**: 再生速度を MIDI ノート番号で指定する制御入力。サンプルの **Root Key** と同じであれば、そのサンプルの本来の音高で再生されます。さらに、**Sel** 入力が開放であれば、サンプル・マップ中のどのサンプルを使って再生するか選択するためにも、この **P** 入力を使います。発振器モードの場合は、生成される信号の基底ピッチがこれで決まります。
- ・ **Sel**: サンプル・マップ中のどのサンプルを使うかを指定するための制御入力。開放の場合は代わりに **P** 入力を基準にサンプルを選択します。
- ・ **F**: 再生速度の変調信号入力。発振器系モジュールの周波数変調と同じ効果です。負の値を与えると再生速度が落ち、値を小さくしていくとついには逆方向に再生されるようになります。
- ・ **St**: トリガー入力にイベントを与えたときの再生開始位置を指定する制御入力。サンプル先頭からのミリ秒単位で指定します。
- ・ **LS**: ループ開始位置を指定する制御入力。サンプル先頭を基準としたミリ秒単位で指定します。開放であればサンプル・ファイル自身に設

定されたループ開始位置を使い、それも未設定ならばサンプルの先頭位置使います。**LS** の値を変えた場合、次にトリガーがかかるか、現在再生中のループの末尾に到達した後で有効になります。

- ・ **LL**: ループ長を指定する制御入力。ミリ秒単位で指定します。開放であればサンプル・ファイル自身に設定されたループ長を使い、それも未設定ならばサンプル全体の長さとしします。**LL** の値を変えた場合、次にトリガーがかかるか、現在再生中のループの末尾に到達した後で有効になります。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **L**: 左チャンネルのオーディオ出力。サンプルがモノラルであるか、**No Stereo** プロパティーがオンであれば、**R** 出力と同じ信号になります。
- ・ **R**: 右チャンネルのオーディオ出力。サンプルがモノラルであるか、**No Stereo** プロパティーがオンであれば、**L** 出力と同じ信号になります。
- ・ **Lng**: サンプル長 (ミリ秒単位) を表す多声イベント出力。

Grain Resynth (グレイン・リシンズ)

サンプラー系



リアル・タイム・リシンセサイザー。モノラル / ステレオのサンプルやサンプル・マップを多声で再生でき、移調も可能です。ピッチと再生速度を独立に制御できるなど、高度なサンプル再生処理ができるようになっています。

一般的なハードウェア・サンプラーや、**Sampler**、**Sampler FM**、**Sampler Loop** などといったモジュールでは、声部ごとにポインターで再生位置を管理しています。したがって出力信号の振幅は、サンプル中、ポインターで示される位置の振幅に常に一致します。このポインターがサンプル上のある速度で移動するため、出力信号の振幅も時間とともに周期的に変化するのが普通です。

ポインターの移動速度により、サンプルの再生速度が決まると同時に、耳に聴こえるピッチも影響を受けます。移動速度を遅くすると、1 周期当たりの時間が長くなる、すなわち周波数 (ピッチ) が低くなるのです。どんどん遅くしていったらポインターが止まると、出力信号の振幅は変化せず、何も音が聴こえなくなります。

このモジュールにも同様にポインターがあります。しかし出力信号の振幅は、サンプル中、ポインターで示される位置の振幅で単純に決まるわけではありません。モジュール内部にはシンセサイザーがあり、ポインター位置の信号をもとに再合成した信号を出力している、というのが真相です。したがってピッチはポインターの移動速度に影響されません。それどころか、ポインターが止まっても、音を生成し続けることが可能です。他のサンプラー系モジュールと同様、ピッチは **P** 入力で制御しますが、これとは別にポインター速度を制御する **Sp** 入力があります。

もちろん、移動速度を落としたり 0 にしたりしただけで、想定通りの音が得られるとは限りません。金槌で釘を打つ音のサンプルは、速度を落とすと釘を打っているようには聴こえなくなってしまいます。**Grain Resynth** の音声再合成アルゴリズムは、こういった音をはじめとするさまざまな音声信号を、時には繊細に、時には劇的に変化させることができます。アルゴリズムの動作は、**G**(粒度、Granularity) および **Sm**(滑らかさ、Smoothness) というパラメーターで調整可能です。また、**Signal-Informed Granulation** プロパティーは、サンプルの読み込み時にサンプル自体に保持されている情報を調べ、これも考慮に入れて再合成するかどうか、を表します。サンプル・マップに属するサンプルごとに、個別に設定できます。オーディオ素材の特性を生かして再合成したい場合はオンにするとよいでしょう。逆にオフにすると、いかにも「電子音楽」的な音になってしまいがちです。

サンプルの管理にはサンプル・マップ・エディターを使います。

正值のゲート・イベントを与えると、**St** 入力で指定された開始位置からサンプルの再生が始まります。**Loop** プロパティーをオンにすると、再生が始まった後、所定のループ範囲に入るとその範囲を繰り返し再生

するようになります。このループ範囲は、サンプル・ファイル中にあらかじめ設定されていますが、未設定の場合はサンプル全体がループ範囲であるとみなします。但し、**LS**、**LL** 入力で、ループ開始位置とループ長を変更することも可能です。一方、**Loop** プロパティをオフにすると、トリガー入力に正值のイベントを与えたとき、先頭から末尾まで一度だけ再生するようになります。但し、**Direction** プロパティが **Backward** であれば、末尾から先頭に向かって逆方向に再生します。

Loop in Release プロパティをオフにすると、負値のゲート・イベントが与えられたとき、ループ範囲の末尾（逆方向再生ならば先頭）に到達した時点でフェード・アウトするようになります。一方、**Loop in Release** プロパティがオンの場合は、負値のゲート・イベントがあってもほとんど影響がありません。**Loop** プロパティをオフにした場合も同様です。

No Stereo プロパティをオンにすると、ステレオ・サンプルであってもモノラル扱い、すなわち、左チャンネルの信号を **L** と **R** の両方に出力するようになります。CPU に対する負荷が問題になる場合、これをオンにすることにより回避できるかも知れません。

G 以外の入力はオーディオ入力になっており、その値はサンプルの再トリガー時（正值のゲート・イベントが与えられたとき）に適用されます。サンプルの再生中、**Gr** 入力で指定された時間間隔ごとに、値が変化していないかどうか調べるようになっています。

- ・ **G**: この入力に正值のゲート・イベントを与えると、**St** 入力で指定された開始位置から再生が始まります。負値のイベントを与えると、ループ末尾に到達した時点で再生が中断します (**Loop in Release** プロパティがオフの場合)。
- ・ **P**: 再生速度を MIDI ノート番号で指定するオーディオ制御入力。サンプルの **Root Key** と同じであれば、そのサンプルの本来の音高で再生されます。さらに、**Sel** 入力が開放であれば、サンプル・マップ中のどのサンプルを使って再生するか選択するためにも、この **P** 入力を使います。
- ・ **Sel**: サンプル・マップ中のどのサンプルを使うかを指定するためのオーディオ制御入力。開放の場合は代わりに **P** 入力を基準にサンプルを選択します。
- ・ **St**: トリガー入力にイベントを与えたときの再生開始位置を指定する制御入力。サンプル先頭からのミリ秒単位で指定します。

- ・ **LS**: ループ開始位置を指定するオーディオ制御入力。サンプル先頭を基準としたミリ秒単位で指定します。開放であればサンプル・ファイル自身に設定されたループ開始位置を使い、それも未設定ならばサンプルの先頭とします。
- ・ **LL**: ループ長を指定するオーディオ制御入力。ミリ秒単位で指定します。開放であればサンプル・ファイル自身に設定されたループ長を使い、それも未設定ならばサンプル全体の長さとして、**LL=0**であれば、ループ開始位置に到達した時点で再生が停まります。
- ・ **Sp**: ピッチと独立に、出力速度を指定するためのオーディオ制御入力。サンプル本来の速度を基準とした比率で指定します。したがって **Sp=1** ならばサンプル本来の速度、**Sp=2** ならば倍速、**Sp=0** ならば停止、ということになります。この入力開放であれば、一般的なサンプラーと同様に、再生速度に応じてピッチが変化します。
- ・ **Gr**: 再合成処理の粒度をミリ秒単位で指定するオーディオ制御入力。再合成を行う、時間軸方向の単位を表します。**Signal-Informed Granulation** プロパティーがオンであれば、この値そのままではなく、サンプル自体に保持されている情報をもとに調整して使います。
- ・ **SO**: サンプル位置 (Sample Offset) をミリ秒単位で変調するためのオーディオ制御入力。サンプル位置を変えてもピッチには影響がありません。**Noise** モジュールの信号を与えてランダムに再生位置を変える、などの使い方があります。
- ・ **Sm**: 再合成処理の滑らかさを指定するオーディオ制御入力。再合成された信号単位を、時間軸に沿って滑らかに変化させる度合いを表します。値を小さくするほど粗雑な響きになります。
- ・ **Pan**: ステレオ場における定位を指定するオーディオ制御入力。**-1** は左、**0** は中央、**1** は右を表します。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **L**: 左チャンネルのオーディオ出力。
- ・ **R**: 右チャンネルのオーディオ出力。
- ・ **Lng**: サンプル長 (ミリ秒単位) を表す多声イベント出力。
- ・ **Pos**: 現在の再生位置 (ミリ秒単位) を与える多声イベント出力。**Gr** で指定される時間間隔でイベントを出力します。

Grain Pitch Former (グレイン・ピッチ・フォーマー) サンプラー系



リアル・タイム・リシンセサイザー。モノラル / ステレオのサンプルやサンプル・マップ、波形セットを多声で再生できます。波形セットを読み込み、再生できる点に特徴があります。サンプル本来のピッチとは無関係に、自由にピッチを調整できます。再生速度と独立に制御できるほか、スペクトルも適切に調整できるため、ピッチを変えても音色が変化してしまったりありません。

一般的なハードウェア・サンプラーや、**Sampler**、**Sampler FM**、**Sampler Loop** などといったモジュールでは、声部ごとにポインターで再生位置を管理しています。したがって出力信号の振幅は、サンプル中、ポインターで示される位置の振幅に常に一致します。このポインターがサンプル上をある速度で移動するため、出力信号の振幅も時間とともに周期的に変化するのが特徴です。

ポインターの移動速度により、サンプルの再生速度が決まると同時に、耳に聴こえるピッチも影響を受けます。移動速度を遅くすると、1 周期当たりの時間が長くなる、すなわち周波数 (ピッチ) が低くなるのです。どんどん遅くしていったらポインターが止まると、出力信号の振幅は変化せず、何も音が聴こえなくなります。

このモジュールにも同様にポインターがあります。しかし出力信号の振幅は、サンプル中、ポインターで示される位置の振幅で単純に決まるわけではありません。モジュール内部にはシンセサイザーがあり、ポインター位置の信号をもとに再合成した信号を出力している、というのが

真相です。したがってピッチはポインターの移動速度に影響されません。それどころか、ポインターが止まっても、音を生成し続けることが可能です。他のサンプラー系モジュールと同様、ピッチは **P** 入力で制御しますが、これとは別にポインター速度を制御する **Sp** 入力があります。

一般的なハードウェア・サンプラーや **Grain Resynth** モジュールは、サンプル本来のピッチを基準に、これを移調することにより相対的にピッチを変えていました。これに対して **Grain Pitch Former** モジュールは、元のピッチとは無関係に、あるピッチの信号を再合成するようになっています。この意味で、むしろ発振器系モジュールに分類してもよい位かも知れません。元の音声素材の性質やモジュール設定にもよりますが、いかにも「電子音楽」的な響きになる傾向があります。さらに、シンバルやゴングのように音程感のないサンプルを処理して、音程を与えることすら可能です。とは言え、サンプルの基底ピッチが明確で、そこからあまり離れない信号を合成する方が、もちろん本来の響きに近い結果が得られます。

また、一般的なハードウェア・サンプラーや **Grain Resynth** モジュールは、基底ピッチの移調に伴い、個々のスペクトルの周波数もずれてしまっていました。これにより響きが変わってしまうのは、サンプラー方式で音声を合成する以上避けられない制限であると考えられてきました。人声の響きは基音成分よりもむしろフォルマントによって決まる要素が大きいのですが、ピッチを変換するとそのピッチも変化するため、いわゆる「ミッキー・マウス効果」などが起こってしまうのです。**Grain Pitch Former** はこれを多少ながらも解消するため、基底成分とフォルマントを分離し、フォルマントのピッチをずらす量を **FS** として独立に制御できるようにしています。人の耳はあらゆるスペクトル要素を用いて基底ピッチを識別しているので、**FS** の値を調整するだけでも、基底ピッチや音色が変わって聴こえます。この効果は特に低音の場合に顕著です。なお、同様の効果を、周波数の異なる2つの発振器を同期させる、すなわち、一方の発振信号の周期ごとにもう一方の信号の位相を強制リセットする、「オシレーター・シンク」という方法で実現している人もいます。

サンプルの管理にはサンプル・マップ・エディターを使います。

正值のゲート・イベントを与えると、**St** 入力で指定された開始位置からサンプルの再生が始まります。**Loop** プロパティをオンにすると、再生が始まった後、所定のループ範囲に入るとその範囲を繰り返し再生するようになります。このループ範囲は、サンプル・ファイル中にあら

はじめ設定されていますが、未設定の場合はサンプル全体がループ範囲であるとみなします。但し、**LS**、**LL** 入力で、ループ開始位置とループ長を変更することも可能です。一方、**Loop** プロパティをオフにすると、トリガー入力に正值のイベントを与えたとき、先頭から末尾まで一度だけ再生するようになります。但し、**Direction** プロパティが **Backward** であれば、末尾から先頭に向かって逆方向に再生します。

Loop in Release プロパティをオフにすると、負値のゲート・イベントが与えられたとき、ループ範囲の末尾（逆方向再生ならば先頭）に到達した時点でフェード・アウトするようになります。一方、**Loop in Release** プロパティがオンの場合は、負値のゲート・イベントがあってもほとんど影響がありません。**Loop** プロパティをオフにした場合も同様です。

No Stereo プロパティをオンにすると、ステレオ・サンプルであってもモノラル扱い、すなわち、左チャンネルの信号を **L** と **R** の両方に出力するようになります。CPU に対する負荷が問題になる場合、これをオンにすることにより回避できるかも知れません。

G 以外の入力はオーディオ入力になっており、その値はサンプルの再トリガー時（正值のゲート・イベントが与えられたとき）に適用されます。サンプルの再生中、**P** 入力で指定された時間間隔ごとに、値が変化していないかどうか調べるようになっています。

- ・ **G**: この入力に正值のゲート・イベントを与えると、**St** 入力で指定された開始位置から再生が始まります。負値のイベントを与えると、ループ末尾に到達した時点で再生が中断します (**Loop in Release** プロパティがオフの場合)。
- ・ **P**: 再生速度を MIDI ノート番号で指定するオーディオ制御入力。**Sel** 入力が開放であれば、サンプル・マップ中のどのサンプルを使って再生するか選択するためにも、この **P** 入力を使います。
- ・ **Sel**: サンプル・マップ中のどのサンプルを使うかを指定するためのオーディオ制御入力。開放の場合は代わりに **P** 入力を基準にサンプルを選択します。
- ・ **St**: トリガー入力にイベントを与えたときの再生開始位置を指定する制御入力。サンプル先頭からのミリ秒単位で指定します。
- ・ **LS**: ループ開始位置を指定するオーディオ制御入力。サンプル先頭を基準としたミリ秒単位で指定します。開放であればサンプル・ファイル自身に設定されたループ開始位置を使い、それも未設定ならばサンプルの先頭とします。

- ・ **LL**: ループ長を指定するオーディオ制御入力。ミリ秒単位で指定します。開放であればサンプル・ファイル自身に設定されたループ長を使い、それも未設定ならばサンプル全体の長さとしています。**LL=0**であれば、ループ開始位置に到達した時点で再生が停まります。
- ・ **Sp**: ピッチと独立に、出力速度を指定するためのオーディオ制御入力。サンプル本来の速度を基準とした比率で指定します。したがって **Sp=1** ならばサンプル本来の速度、**Sp=2** ならば倍速、**Sp=0** ならば停止、ということになります。この入力開放であれば、一般的なサンプラーと同様に、再生速度に応じてピッチが変化します。
- ・ **FS**: ピッチとは独立にフォルマント位置をずらす量 (半音単位) を指定するオーディオ制御入力。
- ・ **SO**: サンプル位置 (Sample Offset) をミリ秒単位で変調するためのオーディオ制御入力。サンプル位置を変えてもピッチには影響がありません。**Noise** モジュールの信号を与えてランダムに再生位置を変える、などの使い方があります。
- ・ **Sm**: 再合成処理の滑らかさを指定するオーディオ制御入力。再合成された信号単位を、時間軸に沿って滑らかに変化させる度合いを表します。値を小さくするほど粗雑な響きになります。
- ・ **Pan**: ステレオ場における定位を指定するオーディオ制御入力。**-1** は左、**0** は中央、**1** は右を表します。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **L**: 左チャンネルのオーディオ出力。
- ・ **R**: 右チャンネルのオーディオ出力。
- ・ **Lng**: サンプル長 (ミリ秒単位) を表す多声イベント出力。
- ・ **Pos**: 現在の再生位置 (ミリ秒単位) を与える多声イベント出力。基底ピッチの周期に相当する時間間隔でイベントを出力します。



複数のステレオ・サンプルを用いるグラニューラー・シンセサイザー。グレインごとに、ピッチ **P**、ピッチ・スライド **PS**、用いるサンプルの選択 **Sel**、サンプル上の位置 **Pos** および長さ **Len** を、独立に制御できます。各グレインのエンベロープについては、アタック時間 **Att** とデケイ時間 **Dec** を制御できます。

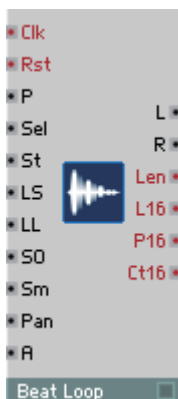
各グレインについて、次のグレインを開始するまでのデルタ時間を、**Dist** として設定できます。同時に使うグレインの最大数はプロパティーとして設定します。

また、ジッター (ある値のゆらぎ) に関する入力はいくつかあり、その値の範囲も設定できます。

サンプルの管理にはサンプル・マップ・エディターを使います。

- ・ **Trig**: ゲート信号を与えるイベント入力。正値のイベントを与えると直ちに次のグレインが始まります。
- ・ **P**: ピッチを MIDI ノート番号 (0 ~ 127) で指定するオーディオ入力。サンプル再生速度とは独立に制御できます。サンプルの **Root Key** と同じであれば、そのサンプルの本来の音高で再生されます。デフォルト値は 60。

- ・ **D/F**: **P** が開放でない場合は再生方向 (D)、開放の場合は周波数 (F) を制御するオーディオ入力。**F**=1 ならばサンプルの本来のピッチ、**F**=-1 ならば逆方向で再生されます。値の範囲は -4 ~ 4、デフォルト値は 1。
- ・ **PJ**: ピッチのゆらぎを半音単位で制御するオーディオ入力 (0 ~ 3)。
- ・ **PS**: グレインのピッチをずらす量を半音単位で制御するオーディオ入力 (-3 ~ 3)。
- ・ **Sel**: 複数のサンプルのうちどれを使うかを、MIDI ノート番号 (0 ~ 127) で選択するオーディオ入力。開放の場合は **P** の値を援用します。
- ・ **Pos**: サウンド・ファイル上の再生位置をミリ秒単位 (0 ~ サンプル長) で指定するオーディオ入力。
- ・ **PsJ**: サウンド・ファイル上の再生位置のゆらぎをミリ秒単位 (0 ~ サンプル長) で指定するオーディオ入力。
- ・ **Len**: グレイン長をミリ秒単位 (10 ~ 100) で指定するオーディオ入力。デフォルト値は 20 ミリ秒。
- ・ **LnJ**: グレイン長のゆらぎをミリ秒単位 (10 ~ 100) で指定するオーディオ入力。デフォルト値は 0 ミリ秒。
- ・ **Att**: アタック時間 (0 ~ 1) を設定するオーディオ入力。デフォルト値は 0.2。
- ・ **Dec**: デケイ時間 (0 ~ 1) を設定するオーディオ入力。デフォルト値は 0.2。
- ・ **Dist**: 次のグレインを開始するまでのデルタ時間をミリ秒単位 (5 ~ 100) で設定するオーディオ入力。デフォルト値は 20 ミリ秒。
- ・ **DisJ**: デルタ時間のゆらぎをミリ秒単位 (10 ~ 100) で設定するオーディオ入力。デフォルト値は 20 ミリ秒。
- ・ **Pan**: ステレオ場における定位を設定するオーディオ入力。最も左が -1、最も右が 1 になります。
- ・ **PnJ**: 定位のゆらぎ (0 ~ 1) を設定するオーディオ入力。
- ・ **A**: 振幅を制御するオーディオ入力。0 ~ 1、デフォルト値は 1。
- ・ **L**: 左チャンネルの多声オーディオ出力。-1 ~ 1 の範囲で変動。
- ・ **R**: 右チャンネルの多声オーディオ出力。-1 ~ 1 の範囲で変動。
- ・ **Lng**: サンプル長をミリ秒単位で表すイベント出力。
- ・ **GTr**: グレイン・トリガーを表す。イベント出力。新しいグレインの開始時に 1、停止時に 0 を出力します。



リアル・タイム・リシンセサイザー。ビート・ループ・サンプルに同期して再生できます。ビート・ループの移調は、再生速度とは独立に **P** 入力で設定できます。同期源として 96 分音符単位のクロックを **C** 入力に与えますが、開放ならば REAKTOR 内蔵の大域クロックに同期します。いずれにしても、サンプル本来の速度とは無関係に、自由に再生速度を調整できます。さらに、シーケンサー系モジュールや MIDI クロックと接続して、リズム感が明瞭なサンプル再生が可能です。

サンプルの管理にはサンプル・マップ・エディターを使います。

Beat Loop モジュールに与えるサンプルは、2、4、8、16、32 分音符などの単位の拍に合わせて正確に区切り、速度を 87 ~ 174BPM に設定する必要があります。この条件が満たされていれば、再生速度を変えても、高い音質で再生可能です。**Pitched Sound** プロパティをオンにすると、通奏低音のピッチがゆらぐのを抑えることができますが、リズムの正確さは多少犠牲になります。

ループ範囲は **LS**、**LL** 入力で設定できます。開放であればサンプル全体がループ範囲であるとみなします。正値のイベントを **Rst** に与えると、再生位置が **St** で指定される位置に戻ります。

No Stereo プロパティをオンにすると、ステレオ・サンプルであってもモノラル扱い、すなわち、左チャンネルの信号を **L** と **R** の両方に出力するようになります。CPU に対する負荷が問題になる場合、これをオンにすることにより回避できるかも知れません。

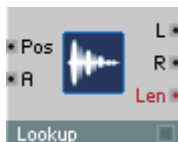
C および **Rst** を除く入力はすべてオーディオ入力になっており、サンプルの再生中、16 分音符相当の時間間隔ごとに、値が変化していかどうか調べるようになっていきます。

- ・ **C**: この入力に与える正值のイベントに同期して、サンプル上の位置が 96 分音符単位で進みます。開放ならば REAKTOR 内蔵の大域クロックに同期します。
- ・ **Rst**: 正值のイベントを与えると、**St** 入力で指定された開始位置から再生が始まります。
- ・ **P**: 再生速度を MIDI ノート番号で指定するオーディオ制御入力。**Sel** 入力開放であれば、サンプル・マップ中のどのサンプルを使って再生するか選択するためにも、この **P** 入力を使います。
- ・ **Sel**: サンプル・マップ中のどのサンプルを使うかを指定するためのオーディオ制御入力。開放の場合は代わりに **P** 入力を基準にサンプルを選択します。
- ・ **St**: トリガー入力にイベントを与えたときの再生開始位置を指定するオーディオ制御入力。サンプル先頭から、16 分音符単位の長さで指定します。
- ・ **LS**: ループ開始位置を指定する制御入力。サンプル先頭を基準として、16 分音符単位の長さで指定します。開放であれば **LS**=0 とみなします。
- ・ **LL**: ループ長を指定する制御入力。16 分音符単位で指定します。開放であればサンプル全体の長さとしします。
- ・ **SO**: サンプル位置 (Sample Offset) を変調するためのオーディオ制御入力。16 分音符単位の長さで指定します。シーケンサー系のモジュールで再生位置を制御するなどの使い方があります。
- ・ **Sm**: 再合成処理の滑らかさを指定するオーディオ制御入力。再合成された信号単位を、時間軸に沿って滑らかに変化させる度合いを表します。値を小さくしていくと、16 分音符相当の時間間隔で炸裂音が聴こえるようになります。
- ・ **Pan**: ステレオ場における定位を指定するオーディオ制御入力。-1 は左、0 は中央、1 は右を表します。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **L**: 左チャンネルのオーディオ出力。
- ・ **R**: 右チャンネルのオーディオ出力。
- ・ **Lng**: サンプル長 (ミリ秒単位) を表す多声イベント出力。
- ・ **L16**: サンプル長 (16 分音符単位) を表す多声イベント出力。

- ・ **P16**: サンプル位置 (16 分音符単位) を表す多声イベント出力。
- ・ **Ct16**: 開始 / リセット以降の経過時間 (16 分音符単位) を表す多声イベント出力。

Sample Lookup (サンプル・ルックアップ)

サンプラー系



サンプルを時間の関数として扱い、表を引くような形で値を求めるモジュール。サンプル上の位置を、ミリ秒単位で **Pos** 入力に与えると、それに対応する値が出力されます。

サンプル・ファイルはコンテキスト・メニューの **Load Sound...** コマンドで読み込みます。

サンプルの管理にはサンプル・マップ・エディターを使います。

音質はプロパティーとして、**Poor**、**Good**、**Excellent** の 3 段階で設定できます。サンプル間を補間する度合いに違いがあり、**Poor** ならばまったく補間しないことになります。但し音質を上げると CPU に対する負荷も高くなります。

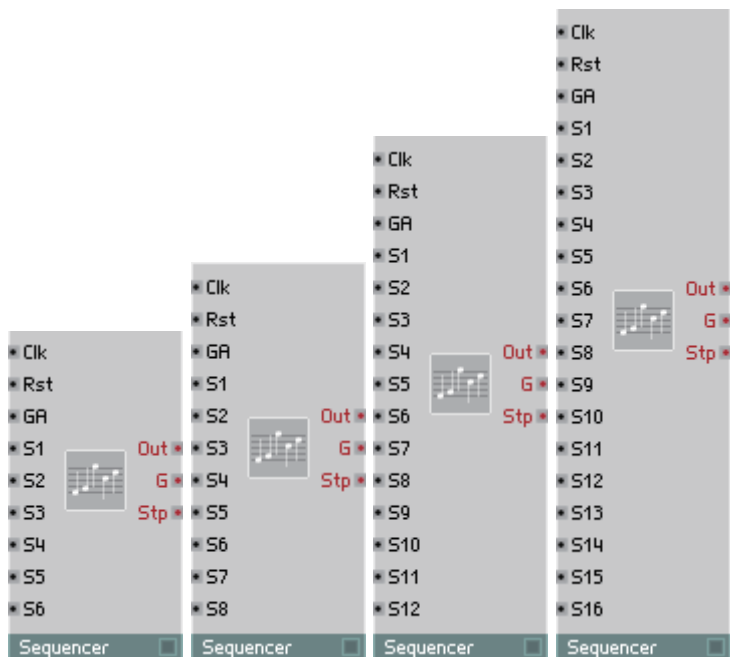
- ・ **Pos**: サンプル上の位置をミリ秒単位で指定するオーディオ入力。
- ・ **A**: 出力信号の振幅を調整するオーディオ入力。
- ・ **L**: 左チャンネルのオーディオ出力。サンプルがモノラルであると、**R** 出力と同じ信号になります。
- ・ **R**: 右チャンネルのオーディオ出力。サンプルがモノラルであると、**L** 出力と同じ信号になります。
- ・ **Lng**: サンプル長 (ミリ秒単位) を表す多声イベント出力。

シーケンサー系モジュール

REAKTOR には、4 種類のステップ数のステップ・シーケンサーと、ステップ選択型シーケンサーがあります。また、これを駆動するためのクロック・モジュールもあります。

Sequencer (シーケンサー)

シーケンサー系



6-Step (6 段ステップ・シーケンサー)

シーケンサー系

6 段のステップ・シーケンサー。各段の出力値 (発振周波数の制御信号) は独立に設定できます。さらに、次の段に進む時点でゲート信号を出力します。この信号の値は、ゲート振幅値入力 **GA** として制御できます。

- ・ **Clk**: クロック信号を与えるオーディオ入力。値が負から正に変わる時点で次の段階に進みます。一般には **Pulse**、**Sync Clock** などのモジュールを接続してください。

- ・ **Rst**: リセット信号を与えるオーディオ入力。値が負から正に変わる時点で先頭段に戻ります。一般には **Button**、**Start/Stop** などのモジュールを接続してください。
- ・ **GA**: ゲート出力信号の振幅を設定するオーディオ入力。値が 0 または開放の場合、ゲート信号は出力されません。
- ・ **S1** ～ **S6**: 各段で出力する値を制御するオーディオ入力。
- ・ **Out**: 各段の信号を出力するイベント出力。
- ・ **G**: ゲート信号を出力するイベント出力。
- ・ **Stp**: 第何段の信号であるかを表すイベント出力。

8-Step (8 段ステップ・シーケンサー) シーケンサー系
段数の違いを除き、機能は **6-Step** モジュールと同様です。

12-Step (12 段ステップ・シーケンサー) シーケンサー系
段数の違いを除き、機能は **6-Step** モジュールと同様です。

16-Step (16 段ステップ・シーケンサー) シーケンサー系
段数の違いを除き、機能は **6-Step** モジュールと同様です。



16 種類の値の中から 1 つを選択、出力するマルチプレクサー。選択信号を一定時間間隔で増やすことにより、シーケンサーとしても使えます。

すなわち、一定時間間隔で増加する信号を **Pos** 入力に与えると、その値を段数 **Len** で割った剰余に当たる番号の値が出力されるので、周期信号が得られることになります。

そのほか、**Pos** 入力には、**Beat Loop** モジュールやランダムなイベント源、マスター・クロックなどを接続できます。

- ・ **Pos**: 段を切り替えるイベント入力。シーケンサーとして使う場合は、16 分音符相当の時間ごとに値を増やしていきます。
- ・ **Len**: 段数の指定 (1 ~ 16)。
- ・ **0 ~ 15**: 各段で出力する値を制御するオーディオ入力。
- ・ **Stp**: 第何段の信号であるかを表すイベント出力。**Pos** を **Len** で割った剰余に相当します。
- ・ **Bar**: 小節番号。**Pos** を **Len** で割った商に相当します。
- ・ **Out**: 各段の信号を出力するイベント出力。

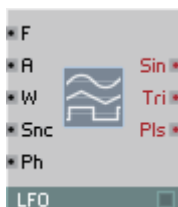
LFO/ エンベロープ系モジュール

LFO/ エンベロープ系には、波形を選択でき周波数変調も可能な LFO(低周波発振器)、ランダム・ステップ波生成器、各種のエンベロープ生成器などのモジュールがあります。時間軸 / レベル軸に関するスロープ波形のエンベロープ生成器も、ステージ数の異なる 3 種類があります。

エンベロープ系モジュールは、**Appearance** ページで **Visible** プロパティをオンにすることにより、波形曲線をパネル上に表示することもできます。表示の大きさは **Size X/Size Y** プロパティで設定します。但し画像の幅を拡大 / 縮小することはできません。

LFO (低周波発振器)

LFO/ エンベロープ系



正弦波、三角波、パルス波の低周波発振器。一般に、変調信号源として、ビブラートやトレモロを実現するために使います。**Settings** ページの **Control Rate** プロパティで指定される周波数のイベント信号を出力します。

制御信号用に特化されているので、オーディオ発振器よりも CPU に対する負荷が小さくて済みます。

- ・ **F**: 発振周波数 (Hz 単位) を指定するオーディオ入力。BPM 単位で指定したい場合は、あらかじめ Hz 単位に変換する必要があります: $F = (1 \text{ 拍当たりの振動数}) \times \text{BPM}/60$ 。例えば 1 小節 (4 拍子) 当たりの振動数が 3 であるとすれば、 $F = (3/4) \times \text{BPM}/60 = 0.0125 \times \text{BPM}$ となります。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **W**: パルス幅を指定するイベント入力 (-1 ~ 1)。0 であれば対称な波形になります。
- ・ **Snc**: 波形の同期信号を与えるオーディオ入力。この信号の立ち上がりで、**Ph** に指定した位相にリセットされます。

- ・ **Ph**: 同期信号の立ち上がりで出力信号がリセットされる位相。1 周期を 1 とした 0 ~ 1 の値で表します。0.5 ならば半周期 (π 、 180°)、1 ならば 1 周期 (2π 、 360°) で実質的に 0 と同じことになります。
- ・ **Sin**: 正弦波のイベント出力。
- ・ **Tri**: 三角波のイベント出力。
- ・ **Pls**: パルスのイベント出力。

Slow Random (スロー・ランダム)

LFO/ エンベロープ系



低周波のランダム・ステップ波発振器。

- ・ **F**: ステップ周波数 (Hz 単位) を指定する制御入力。
- ・ **A**: 出力信号の振幅を調整する制御入力。
- ・ **Out**: ランダム・ステップ波のイベント出力。

H-Env (H-エンベロープ)

LFO/ エンベロープ系



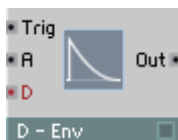
ホールドのパラメーターを備えたエンベロープ生成器。トリガーがかかると **A** で指定されたレベルになり、ホールド時間が経過すると 0 に戻ります。トリガーは、ホールド時間中を含め、いつでもかけることができます。

- ・ **Trig**: 立ち上がりでエンベロープにトリガーをかけるオーディオ入力。
- ・ **A**: 出力の最大振幅レベルを指定するオーディオ入力。トリガーがかかった時点で入力ポートの値をサンプリングします。
- ・ **H**: ホールド時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します ($\text{dB}_{1\text{ms}}$ 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。



ホールドおよびリリースのパラメーターを備えたエンベロープ生成器。トリガーがかかると **A** で指定されたレベルになり、ホールド時間の経過後、リリース時間をかけて指数函数的に減衰し、0 に戻ります。

- ・ **Trig**: 立ち上がりでエンベロープにトリガーをかけるオーディオ入力。
- ・ **A**: 出力の最大振幅レベルを指定するオーディオ入力。トリガーがかかった時点で入力ポートの値をサンプリングします。
- ・ **H**: ホールド時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。



デケイのパラメーターを備えたエンベロープ生成器。トリガーがかかると **A** で指定されたレベルになり、その後、デケイ時間をかけて指数函数的に減衰し、0 に戻ります。

- ・ **Trig**: 立ち上がりでエンベロープにトリガーをかけるオーディオ入力。
- ・ **A**: 出力の最大振幅レベルを指定するオーディオ入力。トリガーがかかった時点で入力ポートの値をサンプリングします。
- ・ **D**: デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。



デケイおよびリリースのパラメーターを備えたエンベロープ生成器。トリガーがかかるとゲート・イベントで指定されたレベルになり、デケイ時間をかけて指数函数的に減衰します。その途中で振幅が0のゲート・イベント(すなわちノート・オフ)を与えると、その時点からリリース時間をかけて減衰し(減衰曲線の係数が変化)、0に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **D**: デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。



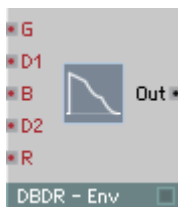
デケイ、サスティン、リリースのパラメーターを備えたエンベロープ生成器。トリガーがかかるとゲート・イベントで指定されたレベルになり、デケイ時間をかけてサスティン・レベルまで指数函数的に減衰します。その途中で振幅が0のゲート・イベント(すなわちノート・オフ)を与えると、その時点からリリース時間をかけて減衰し、0に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。

- ・ **D**: デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **S**: サスティン・レベルを指定するイベント入力。通常の使用範囲は 0 ~ 1 ですが、負の値を含め、この範囲外を指定することも可能です。0 ならば信号レベルが 0 にまで落ち、1 ならばアタック・レベルのまま保たれることを表します。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。

DBDR-Env (DBDR- エンベロープ)

LFO/ エンベロープ系



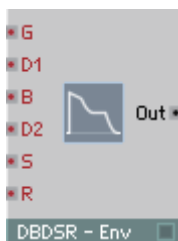
デケイ、ブレイクポイント、デケイ、リリースのパラメーターを備えたエンベロープ生成器。トリガーがかかるとゲート・イベントで指定されたレベルになり、第 1 デケイ時間をかけてブレイクポイント・レベルまで指数函数的に減衰します。次に第 2 デケイ時間をかけて、指数函数的に減衰していきます。その途中で振幅が 0 のゲート・イベント (すなわちノート・オフ) を与えると、その時点からリリース時間をかけて減衰し、0 に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **D1**: 第 1 デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **B**: ブレイクポイント・レベル (振幅の最大値を 1 とする相対値、0 ~ 1) を指定するイベント入力。0 ならば **D2** は意味がありません。1 ならば **D1** は無視して即座に第 2 デケイ段に移行することになります。
- ・ **D2**: 第 2 デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。

- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。

DBDSR-Env (DBDSR-エンベロープ)

LFO/エンベロープ系



デケイ、ブレイクポイント、デケイ、サスティン、リリースのパラメーターを備えたエンベロープ生成器。トリガーがかかるとゲート・イベントで指定されたレベルになり、第 1 デケイ時間をかけてブレイクポイント・レベルまで指数函数的に減衰します。次に第 2 デケイ時間をかけて、サスティン・レベルまで指数函数的に減衰していきます。その途中で振幅が 0 のゲート・イベント (すなわちノート・オフ) を与えると、その時点からリリース時間をかけて減衰し、0 に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **D1**: 第 1 デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **B**: ブレイクポイント・レベル (振幅の最大値を 1 とする相対値、0 ~ 1) を指定するイベント入力。0 ならば **D2** は意味がありません。1 ならば **D1** は無視して即座に第 2 デケイ段に移行することになります。
- ・ **D2**: 第 2 デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **S**: サスティン・レベルを指定するイベント入力。通常の使用範囲は 0 ~ 1 ですが、負の値を含め、この範囲外を指定することも可能です。0 ならば信号レベルが 0 にまで落ち、1 ならばアタック・レベルのまま保たれることを表します。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。

- ・ **Out:** エンベロープ信号のオーディオ出力。

AD-Env (AD-エンベロープ)

LFO/エンベロープ系



アタック、デケイのパラメーターを備えたエンベロープ生成器。トリガーがかかると、アタック時間をかけて最大振幅値まで上昇し、その後すぐにデケイ段に移って、デケイ時間をかけて指数函数的に減衰していきます。

- ・ **Trig:** 立ち上がりでエンベロープにトリガーをかけるオーディオ入力。
- ・ **A:** 出力の最大振幅レベルを指定する制御入力。
- ・ **A:** アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **D:** デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out:** エンベロープ信号のオーディオ出力。

AR-Env (AR-エンベロープ)

LFO/エンベロープ系



アタック、リリースのパラメーターを備えたエンベロープ生成器。ゲート信号によりトリガーがかかると、アタック時間をかけて線形に上昇し、ゲート信号の振幅値により指定されたレベルになってそのままのレベルを保ちます。その後、振幅が0のゲート・イベント (すなわちノート・オフ) を与えると、その時点からリリース時間をかけて減衰し、0に戻ります。

- ・ **G:** エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。

- ・ **A:** アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **R:** リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out:** エンベロープ信号のオーディオ出力。

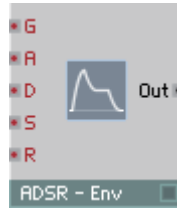
ADR-Env (ADR-エンベロープ)

LFO/エンベロープ系



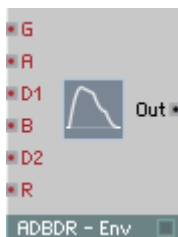
アタック、デケイ、リリースのパラメーターを備えたエンベロープ生成器。ゲート信号によりトリガーがかかると、アタック時間をかけて線形に上昇し、ゲート信号の振幅値により指定されたレベルになります。その後、デケイ時間をかけて減衰します。振幅が0のゲート・イベント(すなわちノート・オフ)を与えると、その時点からリリース時間をかけて指数函数的に減衰し、0に戻ります。

- ・ **G:** エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **A:** アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **D:** デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **R:** リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out:** エンベロープ信号のオーディオ出力。



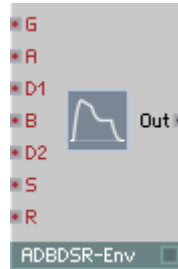
アタック、デケイ、サスティン、リリースのパラメーターを備えたエンベロープ生成器。ゲート信号によりトリガーがかかると、アタック時間をかけて線形に上昇し、ゲート信号の振幅値により指定されたレベルになります。その後、デケイ時間をかけてサスティン・レベルまで減衰します。振幅が0のゲート・イベント(すなわちノート・オフ)を与えると、その時点からリリース時間をかけて指数函数的に減衰し、0に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **A**: アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **D**: デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **S**: サスティン・レベルを指定するイベント入力。通常の使用範囲は0 ~ 1 ですが、負の値を含め、この範囲外を指定することも可能です。0 ならば信号レベルが0 にまで落ち、1 ならばアタック・レベルのまま保たれることを表します。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。



アタック、デケイ、ブレイクポイント、デケイ、リリースのパラメーターを備えたエンベロープ生成器。ゲート信号によりトリガーがかかると、アタック時間をかけて線形に上昇し、ゲート信号の振幅値により指定されたレベルになります。その後、第1デケイ時間をかけてブレイクポイント・レベルまで指数函数的に減衰します。次に第2デケイ時間をかけて、指数函数的に減衰していきます。その途中で振幅が0のゲート・イベント(すなわちノート・オフ)を与えると、その時点からリリース時間をかけて減衰し、0に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **A**: アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **D1**: 第1デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **B**: ブレイクポイント・レベル(振幅の最大値を1とする相対値、0～1)を指定するイベント入力。0 ならば **D2** は意味がありません。1 ならば **D1** は無視して即座に第2デケイ段に移行することになります。
- ・ **D2**: 第2デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。



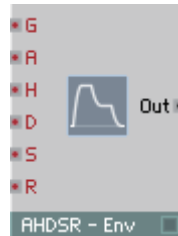
アタック、デケイ、ブレイクポイント、デケイ、サスティン、リリースのパラメーターを備えたエンベロープ生成器。ゲート信号によりトリガーがかかると、アタック時間をかけて線形に上昇し、ゲート信号の振幅値により指定されたレベルになります。その後、第1デケイ時間をかけてブレイクポイント・レベルまで指数函数的に減衰します。次に第2デケイ時間をかけて、サスティン・レベルまで指数函数的に減衰していきます。その途中で振幅が0のゲート・イベント(すなわちノート・オフ)を与えると、その時点からリリース時間をかけて減衰し、0に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **A**: アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します ($\text{dB}_{1\text{ms}}$ 単位)。
- ・ **D1**: 第1デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します ($\text{dB}_{1\text{ms}}$ 単位)。
- ・ **B**: ブレイクポイント・レベル(振幅の最大値を1とする相対値、0～1)を指定するイベント入力。0 ならば **D2** は意味がありません。1 ならば **D1** は無視して即座に第2デケイ段に移行することになります。
- ・ **D2**: 第2デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します ($\text{dB}_{1\text{ms}}$ 単位)。
- ・ **S**: サスティン・レベルを指定するイベント入力。通常の使用範囲は0～1ですが、負の値を含め、この範囲外を指定することも可能です。0 ならば信号レベルが0にまで落ち、1 ならばアタック・レベルのまま保たれることを表します。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します ($\text{dB}_{1\text{ms}}$ 単位)。

- ・ **Out:** エンベロープ信号のオーディオ出力。

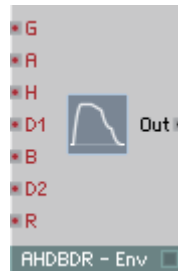
AHDSR-Env (AHDSR-エンベロープ)

LFO/エンベロープ系



アタック、ホールド、デケイ、サスティン、リリースのパラメーターを備えたエンベロープ生成器。ゲート信号によりトリガーがかかると、アタック時間をかけて線形に上昇し、ゲート信号の振幅値により指定されたレベルになります。そのレベルをホールド時間保った後、その後、デケイ時間をかけてサスティン・レベルまで減衰します。振幅が0のゲート・イベント（すなわちノート・オフ）を与えると、その時点からリリース時間をかけて指数函数的に減衰し、0に戻ります。

- ・ **G:** エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **A:** アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **H:** ホールド時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **D:** デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **S:** サスティン・レベルを指定するイベント入力。通常の使用範囲は 0 ~ 1 ですが、負の値を含め、この範囲外を指定することも可能です。0 ならば信号レベルが 0 にまで落ち、1 ならばアタック・レベルのまま保たれることを表します。
- ・ **R:** リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out:** エンベロープ信号のオーディオ出力。



アタック、ホールド、デケイ、ブレイクポイント、デケイ、リリースのパラメーターを備えたエンベロープ生成器。ゲート信号によりトリガーがかかると、アタック時間をかけて線形に上昇し、ゲート信号の振幅値により指定されたレベルになります。そのレベルをホールド時間保った後、第1デケイ時間をかけてブレイクポイント・レベルまで指数函数的に減衰します。次に第2デケイ時間をかけて、指数函数的に減衰していきます。その途中で振幅が0のゲート・イベント(すなわちノート・オフ)を与えると、その時点からリリース時間をかけて減衰し、0に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **A**: アタック時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **H**: ホールド時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **D1**: 第1デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **B**: ブレイクポイント・レベル(振幅の最大値を1とする相対値、0～1)を指定するイベント入力。0 ならば **D2** は意味がありません。1 ならば **D1** は無視して即座に第2デケイ段に移行することになります。
- ・ **D2**: 第2デケイ時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **R**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **Out**: エンベロープ信号のオーディオ出力。



4 段の折れ線で表されるエンベロープの生成器。ゲート信号によりトリガーがかかると、**T1** の時間をかけて **L1** のレベルまで線形に上昇します。次に **T2** の時間をかけて、**L2** のレベルまで線形に上昇または下降します。第 3 段はサステイン段に相当し、**T3** の時間の後もこのレベルを保ちます。振幅が 0 のゲート・イベント (すなわちノート・オフ) を与えると、その時点からリリース時間をかけて減衰し、0 に戻ります。

- ・ **G**: エンベロープのトリガーとして使う、ゲート信号を与えるイベント入力。この振幅によってエンベロープ開始時の出力レベルが決まります。
- ・ **T1/T2/T3**: 各段の時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **L1/L2**: 各段の最後に到達するレベルを指定する制御入力。
- ・ **L3**: サステイン・レベルを指定するイベント入力。通常の使用範囲は 0 ~ 1 ですが、負の値を含め、この範囲外を指定することも可能です。0 ならば信号レベルが 0 にまで落ち、1 ならばアタック・レベルのまま保たれることを表します。
- ・ **TR**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します (dB_{1ms} 単位)。
- ・ **St**: 現在第何段にいるかを表すイベント出力。リリース段の末尾に達した後、新たにトリガーがかかるまでの間は第 0 段です。この信号を使って、他のエンベロープ生成器とチェーン接続する、自分自身に再トリガーをかける、といった使い方ができます。
- ・ **Out**: エンベロープ信号のオーディオ出力。

5-Ramp (5- ランプ)

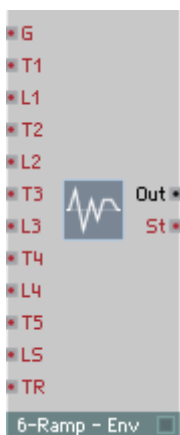
LFO/ エンベロープ系



段数の違いを除き、機能は **4-Ramp** モジュールと同様です。

6-Ramp (6- ランプ)

LFO/ エンベロープ系



段数の違いを除き、機能は **4-Ramp** モジュールと同様です。

フィルター系モジュール

フィルター系は REAKTOR の中でも最も充実したモジュール群です。22 種類のモジュールの中には、標準的なロー・パス、ハイ・パス、バンド・パス・フィルターに加え、Sequential や Moog などといった往年の名機に搭載されたフィルターを再現するものもあります。さらに、残響や散乱などの効果をねらったオール・パス・フィルター（位相フィルター）や、微分器、積分器もあります。

REAKTOR のフィルターはいずれも、0Hz(直流) からサンプル・レートで決まる上限まで、あらゆる周波数帯域で動作します。したがって、オーディオ信号はもちろん、低周波の制御信号に適用して滑らかにする（ポルタメント効果など）使い方も可能です。**P** など、イベント信号しか受け付けられない入力ポートをオーディオ信号で制御したい場合は、**A to E (Perm)** モジュールで変換してください。

フィルターやイコライザーは、周波数特性を制御パネルにグラフ表示できるようになっています。**Appearance** ページで、**Visible** プロパティをオンにしてください。表示の大きさは **Size X**、**Size Y** で指定します。周波数軸は対数目盛で、10Hz から 20kHz の範囲になっています。

HP/LP 1-Pole (ハイ / ロー・パス、1P)

フィルター系



1 ポールのハイ・パス / ロー・パス・フィルター (6dB/oct.)。カットオフ周波数は MIDI ノート番号で指定します。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP**: ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **LP**: ロー・パス・フィルターが施されたオーディオ信号出力。

HP/LP 1-Pole FM (ハイ / ロー・パス、1P、FM)

フィルター系



1 ポールのハイ・パス / ロー・パス・フィルター (6dB/oct.)。カットオフ周波数は MIDI ノート 番号で指定しますが、Hz 単位で増減 (変調) できます。

- ・ **P**: カットオフ周波数を MIDI ノート 番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP**: ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **LP**: ロー・パス・フィルターが施されたオーディオ信号出力。

Allpass 1-Pole (オール・パス、1P)

フィルター系



1 次のオール・パス・フィルター (位相フィルター)。振幅は平坦でどの周波数帯域でも変わりませんが、入力信号と出力信号の位相差は、低周波域の 0 度から高周波域の -180 度まで徐々に変化していきます。位相差が -90 度になるコーナー周波数を **P** で調整できます。

- ・ **P**: コーナー周波数を MIDI ノート 番号 (0 ~ 127) で指定する制御入力。
43 = G1 = 98Hz、84 = C5 = 1047Hz などとなります。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **Out**: フィルター処理が施された信号の出力。

Multi 2-Pole (マルチ、2P)

フィルター系



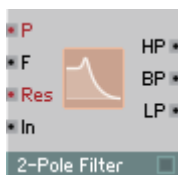
2 ポールのハイ / ロー・パス・フィルター (12dB/oct.) およびバンド・パス・フィルター (6/6dB/oct.)。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定します。

フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP**: ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP**: バンド・パス・フィルターが施されたオーディオ信号出力
- ・ **LP**: ロー・パス・フィルターが施されたオーディオ信号出力。

Multi 2-Pole FM (マルチ、2P、FM)

フィルター系



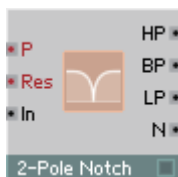
2 ポールのハイ / ロー・パス・フィルター (12dB/oct.) およびバンド・パス・フィルター (6/6dB/oct.)。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP**: ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP**: バンド・パス・フィルターが施されたオーディオ信号出力
- ・ **LP**: ロー・パス・フィルターが施されたオーディオ信号出力。

Multi/Notch 2-Pole (マルチ / ノッチ、2P)

フィルター系



2 ポールのハイ / ロー・パス・フィルター、ノッチ・フィルター (12dB/oct.)、およびバンド・パス・フィルター (6/6dB/oct.)。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定します。

フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

ノッチ・フィルターには、特定の周波数帯域だけを減衰させる働きがあります。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎ

て自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)}$
 $= 1 / (2 - 2 \times \text{Res})$ で求められます。

- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP**: ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP**: バンド・パス・フィルターが施されたオーディオ信号出力
- ・ **LP**: ロー・パス・フィルターが施されたオーディオ信号出力。
- ・ **N**: ノッチ・フィルターが施されたオーディオ信号出力

Multi/Notch 2-Pole FM (マルチ / ノッチ、2P、FM) フィルター系



2 ポールのハイ / ロー・パス・フィルター、ノッチ・フィルター (12dB/oct.)、およびバンド・パス・フィルター (6/6dB/oct.)。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

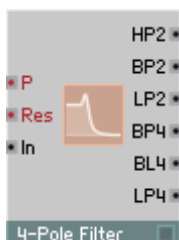
ノッチ・フィルターには、特定の周波数帯域だけを減衰させる働きがあります。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)}$
 $= 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP**: ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP**: バンド・パス・フィルターが施されたオーディオ信号出力
- ・ **LP**: ロー・パス・フィルターが施されたオーディオ信号出力。

- ・ **N**: ノッチ・フィルターが施されたオーディオ信号出力

Multi/LP 4-Pole (マルチ / ロー・パス、4P)

フィルター系



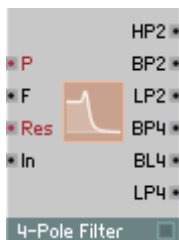
4 ポールの ロー・パス・フィルター (24dB/oct.) を中心に、12/12dB/oct. のバンド・パス、6/18dB/oct. のバンド / ロー・パス、2 ポールのハイ / ロー / バンド・パスの各フィルター出力を組み込んだモジュール。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定します。

フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP2**: 2 ポールのハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP2**: 2 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。
- ・ **LP2**: 2 ポールのロー・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP4**: 4 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。
- ・ **BL4**: 4 ポールのバンド / ロー・パス・フィルターが施されたオーディオ信号出力。

- ・ **LP4**: 4 ポールのロー・パス・フィルターが施されたオーディオ信号出力。

Multi/LP 4-Pole FM (マルチ / ロー・パス、4 ポール、FM フィルター系)



4 ポールのロー・パス・フィルター (24dB/oct.) を中心に、12/12dB/oct. のバンド・パス、6/18dB/oct. のバンド / ロー・パス、2 ポールのハイ / ロー / バンド・パスの各フィルター出力を組み込んだモジュール。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

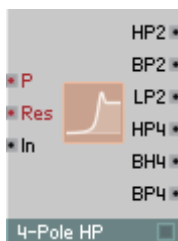
フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP2**: 2 ポールのハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP2**: 2 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。
- ・ **LP2**: 2 ポールのロー・パス・フィルターが施されたオーディオ信号出力。

- ・ **BP4**: 4 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。
- ・ **BL4**: 4 ポールのバンド / ロー・パス・フィルターが施されたオーディオ信号出力。
- ・ **LP4**: 4 ポールのロー・パス・フィルターが施されたオーディオ信号出力。

Multi/HP 4-Pole (マルチ / ハイ・パス、4P)

フィルター系



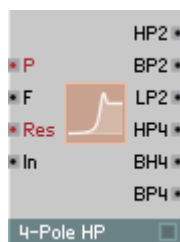
4 ポールの ハイ・パス・フィルター (24dB/oct.) を中心に、12/12dB/oct. のバンド・パス、18/6dB/oct. のバンド / ハイ・パス、2 ポールのハイ / ロー / バンド・パスの各フィルター出力を組み込んだモジュール。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定します。

フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **HP2**: 2 ポールのハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP2**: 2 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。

- ・ **LP2**: 2 ポールのロー・パス・フィルターが施されたオーディオ信号出力。
- ・ **HP4**: 4 ポールのハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BH4**: 4 ポールのバンド / ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP4**: 4 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。

Multi/HP 4-Pole FM (マルチ / ハイ・パス、4P、FM) フィルター系



4 ポールの ハイ・パス・フィルター (24dB/oct.) を中心に、12/12dB/oct. のバンド・パス、18/6dB/oct. のバンド / ハイ・パス、2 ポールのハイ / ロー / バンド・パスの各フィルター出力を組み込んだモジュール。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

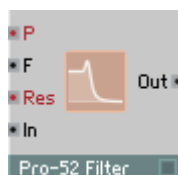
フィルターを通過する帯域の利得は常に 1 (0dB) ですが、カットオフ周波数付近は **Res** の調整により強調することができます。1 に近い値を **Res** に与えると振幅が非常に大きくなるので注意してください。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。

- ・ **HP2**: 2 ポールのハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP2**: 2 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。
- ・ **LP2**: 2 ポールのロー・パス・フィルターが施されたオーディオ信号出力。
- ・ **HP4**: 4 ポールのハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BH4**: 4 ポールのバンド / ハイ・パス・フィルターが施されたオーディオ信号出力。
- ・ **BP4**: 4 ポールのバンド・パス・フィルターが施されたオーディオ信号出力。

Pro-52 Filter (Pro-52 フィルター)

フィルター系



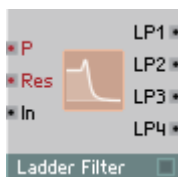
Pro-52 仮想アナログ・シンセサイザーに使われていたフィルターを再現したもの。4 ポール (24dB/oct.) のロー・パス・フィルターを備えています。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

レゾナンス値を 1 に近い値にすると自己発振を起こし、その振幅は約 1 になります。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **In**: フィルター処理を施すオーディオ信号入力。
- ・ **Out**: フィルター処理が施された信号のオーディオ出力。

Ladder Filter (ラダー・フィルター)

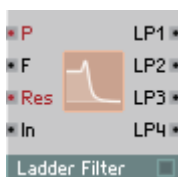
フィルター系



次に挙げる **Ladder Filter FM** と同様ですが、カットオフ周波数の変調機能がありません。

Ladder Filter FM (ラダー・フィルター、FM)

フィルター系



Robert Moog 氏が特許を持ち、ムーグ・シンセサイザーに採用したことで名高いラダー回路をもとに設計したフィルター。24、18、12、6dB/oct. の、4 種類のロー・パス出力を備えた 4 ポール・フィルターです。レゾナンスが可変であり、カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

アナログ回路の飽和特性は、**Distortion** プロパティの調整によりシミュレートできます。ディストーション機能が有効であれば、レゾナンス (Res) を 1 以上にすると自己発振します。その振幅は、Res=1 のときに約 1 で、Res を増やすにつれ急速に大きくなっていきます。

また、シミュレーションの質を **Standard**、**High**、**Excellent** の 3 段階から選べ、その違いは特にディストーションの効果に現れます。もちろん質を高くすると CPU に対する負荷も高くなります。

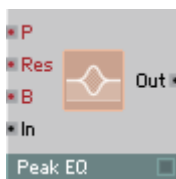
- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。P で決まる周波数を F で微調整します。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)}$

= $1 / (2 - 2 \times \text{Res})$ で求められます。値を 1 以上にすると自己発振によるディストーションの効果が得られます。

- **In:** フィルター処理を施すオーディオ信号入力。
- **LP1:** 6dB/oct. のロー・パス・フィルターが施された信号のオーディオ出力。
- **LP2:** 12/oct. のロー・パス・フィルターが施された信号のオーディオ出力。
- **LP3:** 18dB/oct. のロー・パス・フィルターが施された信号のオーディオ出力。
- **LP4:** 24dB/oct. のロー・パス・フィルターが施された信号のオーディオ出力。

Peak EQ (ピーク・イコライザー)

フィルター系

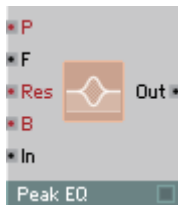


パラメトリック・イコライザー。増幅 / 減衰率、帯域幅、中心 (カットオフ) 周波数を設定できます。中心周波数は MIDI ノート番号で指定します。特定の周波数帯域を増幅 / 減衰する働きがあります。それ以外の帯域には影響を与えません。

- **P:** カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- **Res:** レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- **B:** 増幅 / 減衰率を dB 単位で制御するイベント入力。B=0 ならば信号は変化しません。
- **In:** イコライズ処理を施すオーディオ信号入力。
- **Out:** イコライズ処理が施された信号のオーディオ出力。

Peak EQ FM (ピーク・イコライザー、FM)

フィルター系



パラメトリック・イコライザー。増幅 / 減衰率、帯域幅、中心 (カットオフ) 周波数を設定できます。中心周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。特定の周波数帯域を増幅 / 減衰する働きがあります。それ以外の帯域には影響を与えません。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **Res**: レゾナンスを制御するイベント入力 (0 ~ 1)。0 ならばまったくレゾナンスの効果がありません。逆に 1 に近づけると、効果が強すぎて自己発振します。このときの Q 値は、 $Q = \text{周波数 (Hz)} / \text{帯域幅 (Hz)} = 1 / (2 - 2 \times \text{Res})$ で求められます。
- ・ **B**: 増幅 / 減衰率を dB 単位で制御するイベント入力。 **B**=0 ならば信号は変化しません。
- ・ **In**: イコライズ処理を施すオーディオ信号入力。
- ・ **Out**: イコライズ処理が施された信号のオーディオ出力。

High Shelf EQ (ハイ・シェルフ・イコライザー)

フィルター系

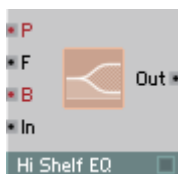


パラメトリック・イコライザー。ハイ・シェルピング特性を持ち、増幅 / 減衰率、帯域幅、カットオフ周波数を設定できます。カットオフ周波数は MIDI ノート番号で指定します。

コーナー周波数以上の帯域を増幅 / 減衰する働きがあります。これ以下の帯域には影響を与えません。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **B**: 増幅 / 減衰率を dB 単位で制御するイベント入力。 **B**=0 ならば信号は変化しません。
- ・ **In**: イコライズ処理を施すオーディオ信号入力。
- ・ **Out**: イコライズ処理が施された信号のオーディオ出力。

High Shelf EQ FM (ハイ・シェルフ・イコライザー、FM) フィルター系



パラメトリック・イコライザー。ハイ・シェルビング特性を持ち、増幅 / 減衰率、帯域幅、カットオフ周波数を設定できます。カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

コーナー周波数以上の帯域を増幅 / 減衰する働きがあります。これ以下の帯域には影響を与えません。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。 **P** で決まる周波数を **F** で微調整します。
- ・ **B**: 増幅 / 減衰率を dB 単位で制御するイベント入力。 **B**=0 ならば信号は変化しません。
- ・ **In**: イコライズ処理を施すオーディオ信号入力。
- ・ **Out**: イコライズ処理が施された信号のオーディオ出力。

Low Shelf EQ (ロー・シェルフ・イコライザー) フィルター系



パラメトリック・イコライザー。ロー・シェルビング特性を持ち、増幅 / 減衰率、帯域幅、カットオフ周波数を設定できます。カットオフ周波数は MIDI ノート番号で指定します。

コーナー周波数以下の帯域を増幅 / 減衰する働きがあります。これ以上の帯域には影響を与えません。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **B**: 増幅 / 減衰率を dB 単位で制御するイベント入力。**B**=0 ならば信号は変化しません。
- ・ **In**: イコライズ処理を施すオーディオ信号入力。
- ・ **Out**: イコライズ処理が施された信号のオーディオ出力。

Low Shelf EQ FM (ロー・シェルフ・イコライザー、FM) フィルター系



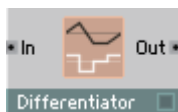
パラメトリック・イコライザー。ロー・シェルビング特性を持ち、増幅 / 減衰率、帯域幅、カットオフ周波数を設定できます。カットオフ周波数は MIDI ノート番号で指定しますが、Hz 単位で増減 (変調) できます。

コーナー周波数以下の帯域を増幅 / 減衰する働きがあります。これ以上の帯域には影響を与えません。

- ・ **P**: カットオフ周波数を MIDI ノート番号 (0 ~ 127) で指定するイベント入力。
- ・ **F**: カットオフ周波数を Hz 単位で指定するオーディオ入力。**P** で決まる周波数を **F** で微調整します。
- ・ **B**: 増幅 / 減衰率を dB 単位で制御するイベント入力。**B**=0 ならば信号は変化しません。
- ・ **In**: イコライズ処理を施すオーディオ信号入力。
- ・ **Out**: イコライズ処理が施された信号のオーディオ出力。

Differentiator (微分器、ディファレンシエーター)

フィルター系



1 ミリ秒ごとに入力信号の変化量を検出し、それに比例した値を出力します。周波数に比例して振幅が変化する、ハイ・パス・フィルターのような働きです。159Hz のとき利得が 0dB になります。

- ・ **In:** フィルター処理を施すオーディオ信号入力。
- ・ **Out:** 微分処理が施された信号のオーディオ出力。

Integrator (積分器、インテグレーター)

フィルター系



積分器。積分定数を初期値として設定できます。1 ミリ秒ごとに入力信号の値を調べ、それに比例した量を累積していきます。周波数に反比例して振幅が変化する、ロー・パス・フィルターのような働きです。159Hz のとき利得が 0dB になります。

- ・ **In:** フィルター処理を施すオーディオ信号入力。
- ・ **Set:** リセット信号を与えるイベント入力。イベントを与えると、その値に出力信号がリセットされます。
- ・ **Out:** 積分処理が施された信号のオーディオ出力。

ディレイ系モジュール

REAKTOR には 6 種類のディレイ系モジュールがあって、一般的な遅延機能はほとんど実現できるようになっています。タップ・ディレイおよびグラニューラー・ディレイがそれぞれ 2 種類のほか、ディフューザー (リバーブ・エフェクトとして有用)、ユニット・ディレイ (ループ・バック処理やモデリング向け) があります。

Single Delay (シングル・ディレイ)

ディレイ系



オーディオ信号、イベント信号のいずれにも適用可能な多声ディレイ・ライン。入力信号を **Dly** 入力で指定された時間遅らせて出力します。

遅延時間の最大値を **Max Delay Buffer** プロパティーで設定できます (デフォルトは 1 秒)。これが RAM の消費量にも影響します。サンプル・レートが 44.1kHz であれば、最大遅延 1 秒あたり、1 声部につき約 172kB を消費します。1 分であれば 10MB 程度です。イベント信号を処理する (**In** 入力の表示が赤) 場合は、**Max Count Of Buffered Events** プロパティーも設定できます。

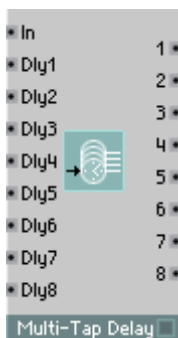
このモジュールは、旧版にあった次のディレイ・モジュールの代替として機能します。

- ・オーディオ信号を **In**、イベント信号を **Dry** に入力すると、REAKTOR 3 で使われていた **Static Delay** モジュールと同等になります。遅延時間がサンプル長の整数倍でなければ、補間によって出力信号を生成します。実際の補間方法はプロパティーとして設定できます。線形に補間すれば高周波成分を若干抑えることができます。
- ・**In** および **Dry** ともにオーディオ信号を入力すると、REAKTOR 3 で使われていた **Modulation Delay** モジュールと同等になります。遅延時間は **Dry** に加えるオーディオ信号によって連続的に変化させることができます。サンプル長の整数倍でない場合の補間については上と同様です。
- ・イベント信号を **In** に入力すると、REAKTOR 3 で使われていた **Event Delay** と同等になります。

- ・ **Dly**: 遅延時間 (ミリ秒単位) を調整するための混成入力。
- ・ **In**: 遅延を生じさせる信号の混成入力。
- ・ **Out**: 遅延を加えた信号の混成出力。

Multi-Tap Delay (マルチ・タップ・ディレイ)

ディレイ系



遅延時間の異なる複数の出力を備えたディレイ・ライン。遅延時間が 1 サンプル相当時間の整数倍でなければ、プロパティとして設定された方法で補間を行います。

出力は通常、ミキサーやスキャナーに接続して使います。

- ・ **In**: 遅延を生じさせる信号のオーディオ入力。
- ・ **Dly1 ~ Dly8**: 遅延時間 (ミリ秒単位) を制御するオーディオ入力。通常の使用範囲: 0 ~ 1000。
- ・ **1 ~ 8: Dly1 ~ Dly8** で指定された分遅れた信号のオーディオ出力。

Diffuser Delay (ディフューザー・ディレイ)

ディレイ系



帰還経路を備えたディレイ・ラインを含むオール・パス・フィルター。オーディオ信号を、特定の周波数帯域を強調することなく、全体的にぼかしたような効果が得られます。リバーブ (残響) 効果を得るための、構成部品としての使い方が典型的でしょう。この場合、**Diffuser Delay** モジュールをいくつか直列につなぎ、少しずつ異なる遅延時間を与えます。

遅延時間は **Dly** 入力で制御しますが、これはオーディオ信号なので連続的に値を変化させることができます。遅延時間がサンプル長の整数倍でなければ、補間によって出力信号を生成します。実際の補間方法はプロパティーとして設定できます。線形に補間すれば高周波成分を若干抑えることができます。

モジュール内部で信号を帰還させる量は、**Dffs** 入力で制御します。

遅延時間を 0 にすると、1 ポールのオール・パス・フィルターと同等の機能になります。これをいくつか直列につなぎ、**Dffs** の値を変調することにより、フェイザーとしての働きをさせることも可能です。

遅延時間の最大値はプロパティーとして設定できます (デフォルトは 200 ミリ秒)。これが RAM の消費量にも影響します。

- ・ **Dly**: 遅延時間 (ミリ秒単位) を調整するための混成入力。
- ・ **Dffs**: 拡散係数を制御するイベント入力 (-1 ~ 1)。0 ならば純粋なディレイ・ラインとして動作し、1 ならば入力と同じ信号、-1 ならば入力を反転した信号になります。通常は 0.5 付近で使います。
- ・ **In**: 遅延を生じさせる信号のオーディオ入力。
- ・ **Out**: 遅延を加えた信号のオーディオ出力。

Grain Delay (グレイン・ディレイ)

ディレイ系



オーディオ信号用のディレイ・ラインですが、ピッチをずらす働きもあります。入力信号を、**Dly** 入力で指定された時間遅らせ、**P** で指定された分だけ移調して出力します。

入力信号を **Gr** で指定される粒度 (ミリ秒) の信号単位に分解して処理します。これを再合成して出力しますが、個々の信号単位を滑らかに変化させる度合いを **Sm** として制御できます。さらに、各信号単位のステレオ場における定位を、**Pan** として設定できます。

遅延時間とピッチは独立に制御できます。**Noise** や **Random** などのモジュールと組み合わせれば面白い効果が得られるでしょう。

再生音質、遅延時間の上限を、プロパティーとして設定できます。但し、実際の遅延時間は、設定された上限よりも 50% 程度長くできます。

- ・ **P**: 移調幅を半音単位で指定するオーディオ入力。
- ・ **Dly**: 遅延時間をミリ秒単位で指定するオーディオ制御入力。
- ・ **Gr**: 再合成処理の粒度をミリ秒単位で指定するオーディオ制御入力。再合成を行う、時間軸方向の単位を表します。
- ・ **Sm**: 再合成処理の滑らかさを指定するオーディオ制御入力。再合成された信号単位を、時間軸に沿って滑らかに変化させる度合いを表します。値を小さくするほど粗雑な響きになります。
- ・ **Pan**: ステレオ場における定位を指定するオーディオ制御入力。-1 は左、0 は中央、1 は右を表します。
- ・ **A**: 振幅を制御するオーディオ入力。
- ・ **In**: 遅延を加えるべきオーディオ信号の入力。
- ・ **L**: 遅延信号の左チャンネルを出力するオーディオ出力。
- ・ **R**: 遅延信号の右チャンネルを出力するオーディオ出力。
- ・ **Dly**: 実際の遅延時間を表す多声イベント出力。信号単位ごとにイベントを出力します。



サンプラー系の **Grain Cloud** モジュールに似ていますが、メモリー中に読み込んだオーディオ・ファイルではなく、モジュールの一番下にある **In** 入力から供給されるオーディオ信号を対象として、オーディオ・バッファー上で適用します。

- ・ **G**: 次のグレインを読み出すためのトリガーを与えるイベント入力。正の値ならばトリガーとなりますが、-1 ならば次のグレインを抑制します (**Dist** 入力も参照)。
- ・ **Frz**: オーディオ・バッファーを凍結するためのイベント入力。正の値を与えると凍結されます。
- ・ **P**: 移調幅を半音単位で指定するオーディオ入力。再生速度とは独立に制御できます。通常の使用範囲は -20 ~ 20 程度です。
- ・ **D/F**: **P** 入力が見線されている場合はグレインを再生する方向、結線されていない場合は周波数を制御するオーディオ入力。**F**=1 ならば、オーディオ・バッファーの中身を順方向で、ピッチを変えずに再生します。**F**=-1 ならば逆方向になります。通常の使用範囲は -4 ~ 4 程度、デフォルト値は 1 です。

- ・ **PJ**: ピッチのゆらぎを半音単位で制御するオーディオ入力。通常の使用範囲: 0 ~ 3。
- ・ **PS**: 現在バッファー上にあるグレインのピッチ・シフトを半音単位で制御するオーディオ入力。通常の使用範囲: -3 ~ 3。
- ・ **Dly**: 遅延時間をミリ秒単位で制御するオーディオ入力。0 からバッファー長までの範囲で指定できます。
- ・ **DIJ**: 遅延時間のゆらぎをミリ秒単位で制御するオーディオ入力。0 からバッファー長までの範囲で指定できます。
- ・ **Len**: グレイン長をミリ秒単位で制御するオーディオ入力。通常の使用範囲: 10 ~ 100。デフォルト値: 30。
- ・ **LnJ**: グレイン長のゆらぎをミリ秒単位で制御するオーディオ入力。通常の使用範囲: 10 ~ 100。デフォルト値: 0。
- ・ **Att**: 各グレインを再生する際のアタック・タイムを制御するオーディオ入力。範囲: 0 ~ 1。デフォルト値: 0.2。
- ・ **Dec**: 各グレインを再生する際のデケイ・タイムを制御するオーディオ入力。範囲: 0 ~ 1。デフォルト値: 0.2。
- ・ **Dist**: 次のグレインを読み込むまでの時間 (デルタ・タイム) をミリ秒単位で制御するオーディオ入力。グレインは自動的にこのレートでトリガーがかかります。通常の使用範囲: 5 ~ 100。デフォルト値: 20。
- ・ **DisJ**: デルタ・タイムのゆらぎをミリ秒単位で制御するオーディオ入力。通常の使用範囲: 5 ~ 100。デフォルト値: 20。
- ・ **Pan**: ステレオ信号の場合にパン (定位) を制御するオーディオ入力。範囲: -1 ~ 1。
- ・ **PnJ**: パン (定位) のゆらぎを制御するオーディオ入力。範囲: -1 ~ 1。
- ・ **A**: 振幅を制御するオーディオ入力。通常の使用範囲: 0 ~ 1。デフォルト値: 1。
- ・ **In**: 遅延を加えるべきオーディオ信号の入力。
- ・ **L**: 左チャンネルの多声オーディオ出力。
- ・ **R**: 右チャンネルの多声オーディオ出力。
- ・ **Dly**: 各グレインの開始時にディレイ時間を出力する多声出力。
- ・ **Gtr**: グレイン・トリガーの多声出力。グレイン開始時に 1、停止時に 0 を出力します。



オーディオ信号に対して 1 サンプル分 (サンプル・レートの逆数) の遅延を加えます。

ある種のフィードバック (帰還) を使うストラクチャーでは、ループ内のどこかにユニット・ディレイを置く必要があります。そのため、明示的にこのモジュールを置かなくても、あるものとして処理するようになっています。

- ・ **In:** 1 サンプル時間分の遅延を加える信号の入力。
- ・ **Out:** 遅延を加えた信号の出力。

歪み系 (ディストーション系) モジュール

オーディオ信号に歪みを加えるディストーション系のモジュールも、シェイパー、クリッパー、ミラーなど多数用意されています。また、スルー・リミッター、ピーク・デテクター、サンプル & ホールド、分周器もあります。

Saturator (サチュレーター)

歪み系



ディストーション・モジュールですが、信号が飽和する付近の特性曲線が滑らかになっています。出力の最大値は ± 2 で、入力の実対値が4以上のときにこうなります。入力の絶対値がごく小さい場合は、波形を変えずにそのまま出力します。

- ・ **In**: サチュレーション処理を施すオーディオ信号の入力。
- ・ **Out**: サチュレーション処理が施されたオーディオ信号出力。

Saturator 2 (サチュレーター 2)

歪み系



4 次の非対称放物型サチュレーターで、特性曲線をさまざまに調整できるようにになっています。

- ・ **LA**: レベルの非対称性。 **LA** = 0 ならば正の信号と負の信号とで飽和レベルは同等です。 **LA** > 0 ならば正の信号に対する飽和レベルが抑制され、 **LA** = 1 ならば 0 になります。逆に **LA** < 0 ならば負の信号に対する飽和レベルが抑制され、 **LA** = -1 ならば 0 になります。
- ・ **KH**: 飽和曲線の二一部分 (屈折部) の硬さ (Knee hardness)。 **KH** = 0 ならば立ち上がりができるだけ柔らかくなります。飽和レベルの最大値までの範囲をすべて使い、丸みのある飽和曲線を得ることができ

ます。**KH** の値を大きくするにつれ、使える範囲は $(1 - \text{KH})$ までと、徐々に制限されるようになります。**KH** = 1 ならば飽和レベルの近傍で信号が急峻にクリップされます。

- **KHA**: ニー部分の硬さの非対称性 (Knee hardness asymmetry)。**KHA** = 0 ならば正の信号と負の信号とでニー部分の硬さは同等です。**KHA** > 0 ならば正の信号に対するニー部分の硬さが減り、**KHA** = 1 ならば 0 になります。逆に **KHA** < 0 ならば負の信号に対するニー部分の硬さが減り、**KHA** = -1 ならば 0 になります。
- **Offs**: 入力信号に対してオフセットとして加えることにより、逆に飽和曲線がシフトしたように働かせるための信号。最後にオフセット分を相殺してから出力します。
- **In**: 歪みを加えるべき信号の入力。
- **Out**: 歪みが加えられた信号の出力。

Clipper (クリッパー)

歪み系



信号レベルをある範囲に制限するクリッパー。上限値 / 下限値を少しでも上回る / 下回るとその値に制限する、ハード・クリッパーです。制限範囲内であれば、何の変化も施さずそのまま通します。

- **Max**: 信号の上限値を制御するオーディオ入力。
- **Min**: 信号の下限値を制御するオーディオ入力。
- **In**: クリップ処理を施すオーディオ信号入力。
- **Out**: クリップ処理が施されたオーディオ信号出力。

Mod. Clipper (モジュレーション・クリッパー)

歪み系



信号レベルをある範囲に制限するクリッパー。値の絶対値が上限値を少しでも上回るとその値に制限する、ハード・クリッパーです。制限範囲内であれば、何の変化も施さずそのまま通します。

- ・ **M:** 信号の上限 / 下限 (絶対値) を制御するオーディオ入力。
- ・ **In:** クリップ処理を施すオーディオ信号入力。
- ・ **Out:** クリップ処理が施されたオーディオ信号出力。

Mirror 1 Level (1 レベル・ミラー)

歪み系



基準レベルを 1 通り設定できるミラー処理。所定のレベルを超えた信号を、そのレベルを基準に反転します。制限範囲内の信号は、何の変化も施さずそのまま通します。

- ・ **Max:** ミラー・レベルを制御するオーディオ入力。
- ・ **In:** ミラー処理を施すオーディオ信号入力。
- ・ **Out:** ミラー処理が施されたオーディオ信号出力。

Mirror 2 Levels (2 レベル・ミラー)

歪み系

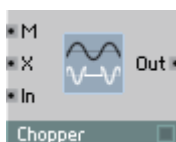


基準レベルを 2 通り設定できるミラー処理。所定のレベルを上回った / 下回った信号を、そのレベルを基準に反転します。制限範囲内の信号は、何の変化も施さずそのまま通します。

- ・ **Max:** ミラー・レベル上限を制御するオーディオ入力。
- ・ **Min:** ミラー・レベル下限を制御するオーディオ入力。
- ・ **In:** ミラー処理を施すオーディオ信号入力。
- ・ **Out:** ミラー処理が施されたオーディオ信号出力。

Chopper (チョッパー)

歪み系



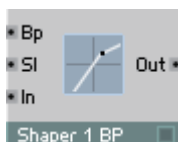
チョッパー。ある条件のときは入力信号の振幅を線形に増幅して出力し、それ以外のときは値 1 の定数値信号を出力します。

条件は変調信号 **M** で切り替えます。これが正であれば入力信号を線形に増幅し、負であれば定数値信号を出力します。

- ・ **M**: 変調信号を与えるオーディオ入力。正か負かの違いだけが動作に影響します。
- ・ **X**: 入力信号の振幅増幅率を制御するオーディオ入力。**M** が正のときにのみ適用されます。
- ・ **In**: チョッパー処理を施すオーディオ信号入力。
- ・ **Out**: チョッパー処理が施されたオーディオ信号出力。

Shaper 1 BP (シェイパー、1 BP)

歪み系



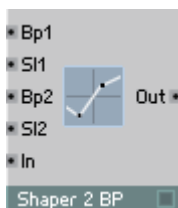
信号レベルを調整するシェイパー。入出力特性曲線は 1 点で区切られた折れ線です。

区切り点より上の直線は、勾配を指定できます。下の直線は傾きが 1 で固定です。

- ・ **Bp**: 区切り点のレベルを制御するオーディオ入力。
- ・ **Sl**: 区切り点より上の直線の勾配を制御するオーディオ入力。
- ・ **In**: シェイピング処理を施すオーディオ信号入力。
- ・ **Out**: シェイピング処理が施されたオーディオ信号出力。

Shaper 2 BP (シェイパー、2 BP)

歪み系



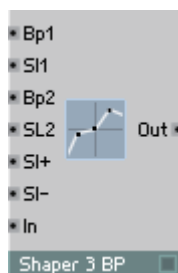
信号レベルを調整するシェイパー。入出力特性曲線は 2 点で区切られた折れ線です。

一方の区切り点より上の直線、もう一方の区切り点より下の直線は、勾配を指定できます。中央部分の直線は傾きが1で固定です。

- ・ **Bp1**: 上の区切り点のレベルを制御するオーディオ入力。
- ・ **SI1**: 上の区切り点より上の直線の勾配を制御するオーディオ入力。
- ・ **Bp2**: 下の区切り点のレベルを制御するオーディオ入力。
- ・ **SI2**: 下の区切り点より下の直線の勾配を制御するオーディオ入力。
- ・ **In**:シェイピング処理を施すオーディオ信号入力。
- ・ **Out**:シェイピング処理が施されたオーディオ信号出力。

Shaper 3 BP (シェイパー、3 BP)

歪み系



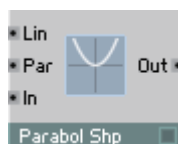
信号レベルを調整するシェイパー。入出力特性曲線は3点(1つは原点に固定)で区切られた折れ線です。

各ブレイクポイント間の勾配を指定できます。

- ・ **Bp1**: 上の区切り点のレベルを制御するオーディオ入力。
- ・ **SI1**: 上の区切り点より上の直線の勾配を制御するオーディオ入力。
- ・ **Bp2**: 下の区切り点のレベルを制御するオーディオ入力。
- ・ **SI2**: 下の区切り点より下の直線の勾配を制御するオーディオ入力。
- ・ **SI+**: 上の区切り点と0の間の直線の勾配を制御するオーディオ入力。
- ・ **SI-**: 下の区切り点と0の間の直線の勾配を制御するオーディオ入力。
- ・ **In**:シェイピング処理を施すオーディオ信号入力。
- ・ **Out**:シェイピング処理が施されたオーディオ信号出力。

Shaper Parabolic (シェイパー、2次の特性曲線)

歪み系



信号レベルを調整するシェイパー。入出力特性曲線を 2 次多項式で設定できます。

特性多項式は、 $\text{Out} = \text{Lin} \times \text{In} + \text{Par} \times \text{In}^2$ となります。

- ・ **Lin**: 特性多項式の 1 次項の係数を制御するオーディオ入力。
- ・ **Par**: 特性多項式の 2 次項の係数を制御するオーディオ入力。
- ・ **In**:シェイピング処理を施すオーディオ信号入力。
- ・ **Out**:シェイピング処理が施されたオーディオ信号出力。

Shaper Cubic (シェイパー、3 次の特性曲線)

歪み系



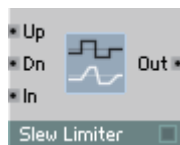
信号レベルを調整するシェイパー。入出力特性曲線を 3 次多項式で設定できます。

特性多項式は、 $\text{Out} = \text{Lin} \times \text{In} + \text{Par} \times \text{In}^2 + \text{Cub} \times \text{In}^3$ となります。

- ・ **Lin**: 特性多項式の 1 次項の係数を制御するオーディオ入力。
- ・ **Par**: 特性多項式の 2 次項の係数を制御するオーディオ入力。
- ・ **Cub**: 特性多項式の 3 次項の係数を制御するオーディオ入力。
- ・ **In**:シェイピング処理を施すオーディオ信号入力。
- ・ **Out**:シェイピング処理が施されたオーディオ信号出力。

Slew Limiter (スルー・リミッター)

歪み系



信号レベル急変時にその勾配を緩和する、スルー・リミッター。勾配の制限値は、上昇時と下降時に別々に設定できます。通常は入力信号をそのまま出力しますが、急変時には一定の勾配で遅れて追従します。

- ・ **Up**: レベル上昇時の最大勾配 (1 秒当たりの最大変化量) を制御するオーディオ入力。

- ・ **Dn**: レベル下降時の最大勾配 (1 秒当たりの最大変化量) を制御するオーディオ入力。
- ・ **In**: 処理対象のオーディオ信号入力。
- ・ **Out**: 処理が施されたオーディオ信号出力。

Peak Detector (ピーク・デテクター)

歪み系



振幅の極大値を検出するモジュール。入力信号レベル (絶対値) が上昇すれば出力値もそれに追従しますが、下降してもそれには追従せず、**Rel** 値で決まる緩い傾斜で滑らかに下降します。アタック時間は 0 です。

その結果、入力信号の振幅包絡線を描く出力が得られるので、エンベロープ・フォロワーとして使えます。

- ・ **Rel**: リリース時間を対数目盛で指定するイベント入力。0 = 1 ms、20 = 10 ms、40 = 100 ms を表します ($\text{dB}_{1\text{ms}}$ 単位)。
- ・ **In**: 処理対象のオーディオ信号入力。
- ・ **Out**: 処理が施されたオーディオ信号出力。

Sample & Hold (サンプル & ホールド)

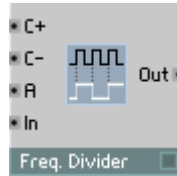
歪み系



サンプル & ホールド。クロックに同期して信号をサンプリングし、次のクロックまで保持します。

クロック信号の立ち上がりで入力値を取得し、そのまま保持します。結果としてステップ波形が得られます。

- ・ **Trig**: クロック信号を与えるオーディオ入力。
- ・ **In**: 処理対象のオーディオ信号入力。
- ・ **Out**: 処理が施されたオーディオ信号出力。



分周器。分周率を High/Low 独立に設定できます。

入力信号のゼロ交叉 (負→正または正→負という変化) を数え、一定数ごとに出力レベル (High/Low) を切り替えることにより、パルス波を生成します。その周波数は $f_{\text{out}} = 2 f_{\text{in}} / (\mathbf{C+} + \mathbf{C-})$ となります。**C+** と **C-** が等しくない場合は、非対称なパルス波になります。

- ・ **C+**: 出力が High であるとき、いくつかのゼロ交叉を検出すると Low に切り替えるか、を制御するオーディオ入力。
- ・ **C-**: 出力が Low であるとき、いくつかのゼロ交叉を検出すると High に切り替えるか、を制御するオーディオ入力。
- ・ **A**: 出力信号の振幅を制御するオーディオ入力。
- ・ **In**: 処理対象のオーディオ信号入力。
- ・ **Out**: 分周されたオーディオ信号出力。

イベント処理系モジュール

イベント処理系モジュールは、計数、論理演算、制御信号の分割や併合、タイミング調整などの目的に使います。また、制御信号の整形、ランダム化に使うモジュールもあります。さらに、機能豊富な検索テーブル・モジュールもあります。これは、発振器系に分類されている **Audio Table** と同等の機能を、制御信号に対して実現するものです。

Accumulator (累算器、アキュムレーター)

イベント処理系



イベント値の累算器。各イベントの値を累算し、合計値を出力します。

- ・ **In**: 累算していく値を与えるイベント入力。
- ・ **Set**: 累算値を初期設定するためのイベント入力。
- ・ **Out**: 合計値を取り出すイベント出力。

Counter (計数器、カウンター)

イベント処理系



イベントの個数の計数器。正值のイベントを **Up** に与えると数が増え、**Dn** に与えると減ります。

- ・ **Up**: 計数するイベントを与える入力。正值のイベントを与えると出力値が増えます。
- ・ **Dn**: 計数するイベントを与える入力。正值のイベントを与えると出力値が減ります。
- ・ **Set**: 計数値を初期設定するためのイベント入力。
- ・ **Out**: イベント数を取り出すイベント出力。

Randomizer (乱数生成器)

イベント処理系



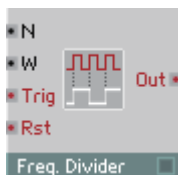
乱数を生成します。数の範囲を指定できます。

入力イベントの値に、ある範囲の乱数を加算して出力します。**Out** = **In** + **X**、ここで **X** の絶対値は **Rng** 以下。

- ・ **Rng**: 乱数の範囲を制御するオーディオ入力。
- ・ **In**: 乱数を加える元となる信号を与えるイベント入力。
- ・ **Out**: 結果を取り出すイベント出力。

Frequency Divider (分周器)

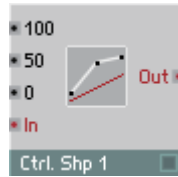
イベント処理系



N 入力で指定された率で入力イベント信号を分周します。すなわち、**N** 個のイベントを数えた時点までを 1 周期とし、デューティ比 **W** のパルス波にして出力します。

- ・ **N**: 分周率を表す制御入力。値が小数の場合は切り捨てて整数にします。
- ・ **W**: 出力信号 (パルス波) のデューティ比を指定する制御入力 (0 ~ 1)。0.5 ならば 1 周期のちょうど半分で 0 に戻ります。1 ならば出力信号は常に 1 のままです。
- ・ **Trig**: イベント (クロック) 信号入力。MIDI クロックなどを入力します。なお、正值のイベントだけが処理の対象です。
- ・ **Rst**: 内部のカウンターをリセットして強制的に同期するためのイベント入力。MIDI 同期に使う場合は MIDI 開始信号を接続してください。なお、正值のイベントを与えたときのみ有効です。
- ・ **Out**: 分周されたイベント信号出力。

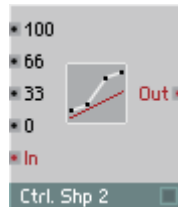
Ctrl. Shaper 1 BP (制御信号用シェイパー、1 BP) イベント処理系



イベント信号のレベルを調整するシェイパー。入出力特性曲線は 1 点で区切られた折れ線です。

- ・ **100:** $In=1$ (100%) のときの出力値。
- ・ **50:** $In=0.5$ (50%、区切り点) のときの出力値。
- ・ **0:** $In=0$ (0%) のときの出力値。
- ・ **In:** シェイピング処理を施すイベント信号入力。
- ・ **Out:** シェイピング処理が施されたイベント信号出力。

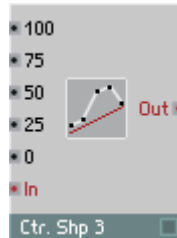
Ctrl. Shaper 2 BP (制御信号用シェイパー、2 BP) イベント処理系



イベント信号のレベルを調整するシェイパー。入出力特性曲線は 2 点で区切られた折れ線です。

- ・ **100:** $In=1$ (100%) のときの出力値。
- ・ **66:** $In=0.66$ (66%、区切り点) のときの出力値。
- ・ **33:** $In=0.33$ (33%、区切り点) のときの出力値。
- ・ **0:** $In=0$ (0%) のときの出力値。
- ・ **In:** シェイピング処理を施すイベント信号入力。
- ・ **Out:** シェイピング処理が施されたイベント信号出力。

Ctrl. Shaper 3 BP (制御信号用シェイパー、3 BP) イベント処理系



イベント信号のレベルを調整するシェイパー。入出力特性曲線は3点で区切られた折れ線です。

- ・ **100:** $In=1$ (100%) のときの出力値。
- ・ **75:** $In=0.75$ (75%、区切り点) のときの出力値。
- ・ **50:** $In=0.5$ (50%、区切り点) のときの出力値。
- ・ **25:** $In=0.25$ (25%、区切り点) のときの出力値。
- ・ **0:** $In=0$ (0%) のときの出力値。
- ・ **In:** シェイピング処理を施すイベント信号入力。
- ・ **Out:** シェイピング処理が施されたイベント信号出力。

Logic AND (論理積) イベント処理系



イベント信号用の論理ゲート。2つの入力信号の論理積を求めます。すなわち、両方とも正值であれば1、それ以外の場合は0を出力します。入力値は、0または負値を偽、正值を真とみなします。

Logic OR (論理和) イベント処理系



イベント信号用の論理ゲート。2つの入力信号の論理和を求めます。すなわち、少なくとも一方が正值であれば1、それ以外の場合は0を出力します。

入力値は、0 または負値を偽、正值を真とみなします。

Logic EXOR（排他的論理和）

イベント処理系



イベント信号用の論理ゲート。2つの入力信号の排他的論理和を求めます。すなわち、一方だけが正值であれば1、それ以外の場合は0を出力します。したがって、一方を正值に固定すれば、論理否定としても使えることになります。

入力値は、0 または負値を偽、正值を真とみなします。

Logic NOT（論理否定）

イベント処理系



イベント信号用の論理ゲート。入力信号の論理否定を求めます。すなわち、入力が正值であれば0、それ以外の場合は1を出力します。

入力値は、0 または負値を偽、正值を真とみなします。

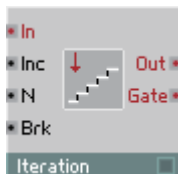
Order（順序制御）

イベント処理系



イベント信号の順序制御。入力イベントに応じ、同じ値のイベントを、**1**、**2**、**3**に順に出力します。**1**出力以降の信号経路にあるモジュールすべての処理が終了してから**2**出力、次いで**3**出力という順序になります。

- ・ **In**: イベント入力。
- ・ **1**: **In**に与えられたイベントと同じ値のイベントを、最初にここから出力。
- ・ **2**: 同じ値のイベントを、2回目にここから出力。
- ・ **3**: 同じ値のイベントを、3回目にここから出力。



In 入力に与えられたイベントは、そのまま出力すると同時に、同じイベントをさらに **N** 回分出力するトリガーとして使います。但しイベントの値は、**Inc** として指定された増分を順次加えて出力します。この N+1 回のイベントはそれぞれ、次のオーディオ・サンプルを処理する前に送信します。パラメーターを増やしながらか処理を反復したい場合に使うとよいでしょう。動作が不安定になりがちなイベント・ループを避ける効果もあります。

- ・ **In**: トリガー入力。このイベントを N+1 回繰り返して出力することになります。
- ・ **Inc**: イベント値の増分。
- ・ **N**: 反復回数。小数第 1 位以下は切り捨てます。

Separator (セパレーター、条件分岐)



入力イベントを、値が閾値より大きいかな否かによって、2 つの出力に分岐させます。

値が正でないイベントを除去することにより (**Thld**=0 とし、**Hi** 出力のみを使う)、ゲート信号をトリガー信号に変換する、といった使い方があります。

- ・ **Thld**: 閾値を指定するオーディオ入力。
- ・ **In**: イベント入力。
- ・ **Hi**: 入力イベントの値が閾値よりも大きい場合のイベント出力。
- ・ **Lo**: 入力イベントの値が閾値よりも小さいか等しい場合のイベント出力。

Value (値変更)

イベント処理系



イベントの値変更モジュール。**In** 入力にイベントが届くと、**Val** に指定された値に変更して出力します。

イベント制御型のサンプル & ホールド機能にも使えます。

- ・ **In**: イベント入力。
- ・ **Val**: イベントの新しい値を指定するイベント入力。
- ・ **Out**: 値を変更したイベントの出力。

Merge (マージ、併合)

イベント処理系



複数のイベント信号を併合します。各入力ポートに信号が与えられた場合に、信号元にかかわらず、最後に受信したイベントを選んで出力します。

入力ポート数は動的に変更できます。**Function** ページの **Min Num Port Groups** プロパティとして指定してください。

旧版と違って、同じ値のイベントが連続しても除去することはありません。必要な場合は **Step Filter** モジュールを使ってください。

Step Filter (ステップ・フィルター)

イベント処理系



直前のイベントを基準として、値が **Tol** で指定された誤差範囲 (正負とも) におさまらない場合にのみ、入力イベントをそのまま出力します。

- ・ **In**: イベントを与える入力。
- ・ **Tol**: 誤差範囲を指定する入力。
- ・ **Out**: イベント出力。

Router M->1 (ルーター M → 1)

イベント処理系



複数系統の入力から 1 系統を選択出力するルーター。**Pos** で指定された番号の入力信号はそのまま出力し、それ以外は廃棄します。

Wrap モードを指定すると、**Pos** の値を (Max + 1) で割った剰余によって信号を選択するようになります。したがって、**Pos** の値として Max + 1 を与えれば 0 を与えたのと同じ、Max + 2 を与えれば 1 を与えたのと同じことになります。

入力ポート数は動的に変更できます。**Function** ページの **Min Num Port Groups** プロパティとして指定してください。

- ・ **Pos**: 出力する信号の番号を与える入力。
- ・ **In**: 信号入力。
- ・ **Out**: 選択された信号の出力。

Router 1,2 (ルーター 1/2)

イベント処理系



1 系統の信号を **Ctrl** 入力によりオン / オフするルーター、または 2 系統の信号を **Ctrl** 入力により切り替えるルーター。オフになっている間は、直前に出力したイベントを保持しています。

入力ポート数は変更可能です。**Function** ページの **Min Num Port Groups** プロパティに、1 または 2 を指定してください。

- ・ **Ctrl**: 信号の切り替えを指示する混成入力。1 系統の場合、正の値であれば入力イベントがそのまま出力されます。2 系統の場合は、正の値であれば 1 番、負または 0 であれば 2 番の信号が出力されます。
- ・ **In**: 切り替えの対象であるイベント入力。
- ・ **Out**: イベント出力。1 系統の場合、**Ctrl** が負の値である間は、直前に出力したイベントがそのまま保持されています。

Router 1→M (ルーター 1 → M)

イベント処理系



1 系統の入力を複数の系統に振り分けるルーター。入力信号を、**Pos** で指定された番号のポートに出力します。

Wrap モードを指定すると、**Pos** の値を $(\text{Max} + 1)$ で割った剰余によって信号を選択するようになります。したがって、**Pos** の値として $\text{Max} + 1$ を与えれば 0 を与えたのと同じ、 $\text{Max} + 2$ を与えれば 1 を与えたのと同じことになります。

出力ポート数は変更可能です。**Function** ページの **Min Num Port Groups** プロパティーで指定してください。

- ・ **Pos**: 出力ポート番号を与える入力。
- ・ **In**: 信号入力。
- ・ **Out**: 信号出力。

Timer (タイマー)

イベント処理系



2 つの入力イベント間の時間を計測し、イベント信号として出力します。また、その逆数を周波数として出力します。

- ・ **In**: 間隔を計測するイベントを与える多声入力。
- ・ **F**: イベント間隔を周期と考え、周波数 (Hz 単位) を出力する多声イベント出力。
- ・ **T**: イベント間隔をミリ秒単位で出力する多声イベント出力。

Hold (ホールド)

イベント処理系



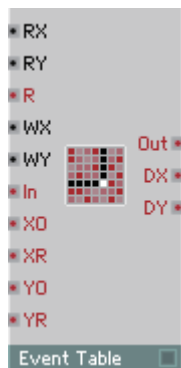
ホールド・エンベロープ生成器。

正值のイベントを与えることによりトリガーをかけると、その値が所定の時間保たれ、その後 0 に戻ります。ホールド時間内に再トリガーをかけることも可能です。

- ・ **Trig**: エンベロープのトリガーとして使うイベント入力。正值のイベントのみが有効です。
- ・ **H**: ホールド時間をミリ秒単位で指定する入力。
- ・ **Out**: エンベロープ信号のイベント出力。

Event Table (イベント・テーブル)

イベント処理系



一連のデータ値を表の形で保持するモジュール。入力信号をインデックスとみなして表を引き、イベント信号として出力します。データ値は、実行時にイベント信号の形で与えるほか、事前に画面上に表示し、編集することも可能です。インデックスは、**X** 入力を使って 1 次元で指定する方法と、**X**、**Y** の両方を使って 2 次元で指定する方法があります。

データ値の読み取りは、**RX** および **RY** にインデックスを与えて行います。逆に実行時にデータ値を与えるときは、値を **IN** に与え、**WX** および **WY** にインデックスを指定します。

X は水平位置 (左から右)、**Y** は垂直位置 (上から下) を表し、番号は 0 から始まります。

制御パネルにはデータ値の一部または全部を表示できます。プロパティの設定により、動作をさまざまに調整できます。

プロパティ、メニュー、キーボード・ショートカットについては、第 22 章「テーブル型のモジュール」を参照してください。

- ・ **RX**: データ値の読み取り用インデックス (X 軸方向) を与えるオーディオ入力。

- ・ **RY**: データ値の読み取り用インデックス (Y 軸方向) を与えるオーディオ入力。データ値を 2 次元で扱う場合に使います。
- ・ **R**: **RX** と **RY** で示されるセルの値を読み込むためのトリガー・イベント入力。その結果、**Out** にイベント値が出力されます。
- ・ **WX**: データ値の書き込み用インデックス (X 軸方向) を与えるオーディオ入力。
- ・ **WY**: データ値の書き込み用インデックス (Y 軸方向) を与えるオーディオ入力。
- ・ **In**: 書き込むデータ値を与えるイベント入力。**W** の値が正になったときに、**WX** と **WY** で示されるセルに書き込まれます。
- ・ **XO**: 制御パネルに表示するインデックス範囲 (X 軸方向) のオフセットを与えるイベント入力。この範囲中のどの位置を表すかは、**View Alignment** プロパティーで設定します。また、単位もプロパティーで設定できます。
- ・ **XR**: 制御パネルに表示するインデックス範囲 (X 軸方向) の長さ、すなわち表示する個数を与えるイベント入力。
- ・ **YO**: 制御パネルに表示するインデックス範囲 (Y 軸方向) のオフセットを与えるイベント入力。この範囲中のどの位置を表すかは、**View Alignment** プロパティーで設定します。また、単位もプロパティーで設定できます。
- ・ **YR**: 制御パネルに表示するインデックス範囲 (Y 軸方向) の長さ、すなわち表示する個数を与えるイベント入力。
- ・ **Out**: **RX** と **RY** で示されるセルの値を出力するオーディオ出力。
- ・ **DX**: X 軸方向の表の大きさを出力するイベント出力。
- ・ **DY**: Y 軸方向の表の大きさを出力するイベント出力。

補助系モジュール

ここまでの節の分類に当てはまらなかったモジュールを集めています。オーディオ・ファイルの録音 / 再生用テープ・デッキをはじめ、声部の管理、オーディオ信号から制御信号への変換、調律やテンポなど REAKTOR の処理に関する状態の表示、といったモジュールがあります。

Tapedeck 1-Ch (1-チャンネル・テープデッキ)

補助系



1-チャンネルのテープデッキで、オーディオ信号の録音、再生が可能です。オーディオ・データの読み書き対象は、プロパティの設定により、メモリーかハード・ディスクかを切り替えることができます。

メモリー・モード

Keep audio only in memory プロパティをオンにするとメモリー・モードになります。再生するオーディオ・データは、**Select File...** ボタンで、あらかじめメモリー中に読み込んでおきます。サンプル・エディターを使うなどして中身を修正した場合は、**Reload** ボタンで読み込み直してください。また、録音するとそのデータはメモリー中に保持されているので、**Save** ボタンを押して保存してください。既に保存済みであれば、既存のファイルに上書きしてよいかどうか訊ねられます。別のファイル名で保存したい場合は **Save as...** ボタンを押してください。

Appearance ページの **Picture** プロパティをオンにしておくと、その波形を制御パネルに表示できます。その際、**Size** プロパティで指定された幅に波形全体が納まるよう、自動的に縮小されます。

最大録音時間 (秒単位) は **Max Recording Size** プロパティで指定しますが、搭載メモリー量によって制限されます。サンプル・レートが 44.1kHz の場合、1 秒分の録音に 86kB、1 分当たり約 5MB が必要です。バッファ・メモリー量の設定が大き過ぎると、録音中に仮想メモリーのスワップが頻繁に起こり、性能を損なう場合があります。

読み込みの際には、そのデータ量に応じたメモリー・バッファールが確保されません。これよりも長く録音する予定であれば、あらかじめ **Max Recording Size** プロパティで録音時間 (秒単位) を指定し、バッファを拡張しておく必要があります。なお、サンプル・レートは、REAKTOR 側で設定した値に自動的に変換されます。

注意：REAKTOR のサンプル・レート設定を変更すると、メモリー中のオーディオ・データすべてについて、変換処理が施されてしまいます。アンサンプル中でこのようなデータを多用しているのであれば、できるだけ避けた方がよいでしょう。なお、ハード・ディスク中にあるオーディオ・データのサンプル・レートを変更した場合は、**Reload** ボタンで読み込み直してください。

ハード・ディスク・モード

Stream audio from/to harddisk プロパティをオンにするとハード・ディスク・モードになります。ハード・ディスクから直接読み込みながら再生し、また、録音の際もデータを直接ハード・ディスクに書き込みます。ファイルは **Select File...** ボタンを押して指定しますが、フォルダーはあらかじめ REAKTOR 全体の環境設定として指定しておきます。**New File** ボタンを押すと、「untitled」という新しいファイルができます (同名のファイルがあれば重複しないよう数字を付加)。但しこの名前は、プロパティ画面上で編集可能です。

読み書きの際、オーディオ・システムの仕様に合わせて、自動的にサンプル・レートを変換します。

Value プロパティをオンにすると、制御パネルにファイル名が表示されるようになります。また、コンテキスト・メニューからファイルの読み書きが可能になります。

- ・ **Rec**: 録音モードのオン / オフを切り替えるゲート信号のイベント入力。正值のイベントを与えると録音が始まります。負値または 0 のイベントを与えるか、最大録音時間に達すると録音が終わります。

- ・ **Play**: 再生モードのオン / オフを切り替えるゲート信号のイベント入力。正值のイベントを与えると再生が始まり、負値または 0 のイベントを与えると停止します。
- ・ **Lp**: ループ再生モードのオン / オフを切り替えるイベント入力。正值のイベントを与えると、末尾まで再生した後再び先頭に戻る、無限ループの状態になります。
- ・ **In**: 録音するオーディオ信号の単声入力。振幅は ± 1 の範囲であるとし、ます。多声信号の録音には **Audio Voice Combiner** モジュールを使ってください。
- ・ **Pse**: 一時停止の制御入力。正值 (通常は 1) ならば一時停止、そうでなければ (通常は 0) 解除します。
- ・ **Pos**: 再生開始位置をミリ秒単位で指定する入力。通常の使用範囲: 0 ~ 20000。
- ・ **Spd**: 再生速度を調整する入力 (正值)。値が 1 ならば本来の速度で再生します。通常の使用範囲: 0.8 ~ 1.2。
- ・ **Out**: 再生 / 録音オーディオ信号の出力。
- ・ **Rec**: 録音中は 1、それ以外は 0 になります。
- ・ **Play**: 再生中は 1、それ以外は 0 になります。
- ・ **Wrp**: ループ再生モードの場合、繰り返しが起こるたびに値が 1 のイベントを送信します。
- ・ **Pse**: 一時停止中は 1、それ以外は 0 になります。
- ・ **Pos**: 再生 / 録音位置をミリ秒単位で出力します。
- ・ **Lng**: 全体の長さをミリ秒単位で出力します。

Tapedeck 2-Ch (2-チャンネル・テープデッキ)

補助系



Tapedeck 1-Ch と同様ですが、オーディオ信号の録音 / 再生用に 2 つのチャンネルがあります。ステレオ・ファイルのインポート / エクスポートが可能です。

- ・ **In L、In R**: 録音するオーディオ信号のうち、左 / 右チャンネルの単声入力。振幅は ± 1 の範囲であるとします。多声信号の録音には **Audio Voice Combiner** モジュールを使ってください。
- ・ **L、R**: 再生オーディオ信号、または録音中のオーディオ信号のうち、左 / 右チャンネルの出力。

Audio Voice Combiner (オーディオ声部合成)

補助系



多声オーディオ信号を合成して単声にします。

- ・ **In**: 多声オーディオ信号の入力。
- ・ **Out**: 合成された単声オーディオ信号の出力。

Event V.C. - All (イベント声部合成 - 全声部)

補助系



多声イベント信号を合成して単声にします。イベントをすべて、1 つの声部の信号に変換して出力します。

- ・ **In**: 多声イベント信号の入力。
- ・ **Out**: 合成された単声イベント信号の出力。

Event V.C. - Max (イベント声部合成 - 最大値の声部)

補助系



多声イベント信号を合成して単声にします。各声部にわたるイベントのうち、値が最大であるものを選んで、1 つの声部の信号として出力します。このモジュールで値を調べる声部を限定するために、**G** 入力を使います。

- ・ **In**: 多声イベント信号の入力。

- ・ **G**: 多声インストゥルメントのゲート信号を与える多声イベント入力。
- ・ **Out**: 合成された単声イベント信号の出力。

Event V.C. - Min (イベント声部合成 - 最小値の声部)

補助系



多声イベント信号を合成して単声にします。各声部にわたるイベントのうち、値が最小であるものを選んで、1つの声部の信号として出力します。このモジュールで値を調べる声部を限定するために、**G** 入力を使います。

- ・ **In**: 多声イベント信号の入力。
- ・ **G**: 多声インストゥルメントのゲート信号を与える多声イベント入力。
- ・ **Out**: 合成された単声イベント信号の出力。

A to E (オーディオ→イベント変換)

補助系



オーディオ信号からイベント信号への変換器。**Settings** メニューの **Control Rate** で指定されたサンプル・レートに従って、オーディオ入力信号をサンプリングし、イベント信号として出力します。

- ・ **A**: オーディオ信号を与える入力。
- ・ **E**: イベント信号出力。

A to E (Trig) (オーディオ→イベント変換、トリガーつき)

補助系



トリガー入力を備えたオーディオ信号からイベント信号への変換器。トリガー信号の立ち上がりに同期してオーディオ信号をサンプリングし、その値をイベント信号として出力します。

- ・ **Trig**: トリガー信号を与えるオーディオ入力。信号の立ち上がりに同期して変換します。

- ・ **A**: オーディオ信号を与える入力。
- ・ **E**: イベント信号出力。

A to E (Perm) (オーディオ→イベント変換、一定周期)

補助系



一定の時間間隔でオーディオ信号をイベント信号に変換する変換器。時間間隔はサンプリング周波数 (Hz 単位) として指定可能です。この周期でオーディオ信号をサンプリングし、その値をイベント信号として出力します。但し周波数が高過ぎる (おおむね 1000Hz 以上) と、CPU が過負荷になる恐れがあります。通常の用途では 200Hz 程度で充分です。

- ・ **F**: サンプリング周波数 (Hz 単位) を指定するオーディオ入力。
- ・ **A**: オーディオ信号を与える入力。
- ・ **E**: イベント信号出力。

A to Gate (オーディオ→ゲート変換)

補助系



オーディオ信号からゲート・イベントへの変換器で、出力の振幅を制御できます。トリガー信号の立ち上がりでゲートが開き、**A** 入力で指定された振幅になります。逆に立ち下がりでゲートが閉じ、振幅は 0 になります。

- ・ **In**: ゲート信号をトリガーするオーディオ入力。
- ・ **A**: ゲート・イベントの振幅を制御するオーディオ入力。
- ・ **G**: ゲート信号のイベント出力。

To Voice (特定声部への転送)

補助系



単声の入力信号を、多声出力信号の、特定の声部に転送する変換器。声部番号は **V** 入力で指定します。有効な声部番号でない場合は、入力信号をそのまま破棄します。

多声のシーケンサーを組み立てる場合などに有用です。

- ・ **V**: 声部番号を指定する単声入力。通常の使用範囲: 1 ~ 10。
- ・ **In**: 転送元の単声信号を与える混成入力。
- ・ **Out**: 多声混成出力。入力信号を、値を変えることなくそのまま、所定の声部に転送します。

From Voice (特定声部からの転送)

補助系

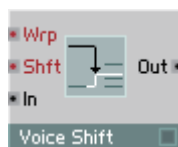


多声の入力信号から、特定の声部のものだけを取り出して転送する変換器。声部番号は **V** 入力で指定します。他の声部の信号はそのまま破棄します。

- ・ **V**: 声部番号を指定する単声入力。通常の使用範囲: 1 ~ 10。
- ・ **In**: 転送元の多声信号を与える混成入力。
- ・ **Out**: 単声混成出力。所定の声部からの入力信号を、値を変えることなくそのまま転送します。

Voice Shift (声部番号のシフト)

補助系



入力信号の声部番号を割り当て直す変換器。例えば声部番号 1/2/3/4 の信号を、声部番号 2/3/4/1、あるいは声部番号 3/4/1/2 に割り当て直すことができます。複数の声部を 1 つの声部に割り当て直す場合は、各声部の信号を混ぜ合わせたものになります。例えば声部番号 1/2 の信号を声部番号 3 に割り当て直すと、声部 3 には、声部 1 と声部 2 の両方の信号がある状態で出力されます。

- ・ **Wrp**: 新しい声部番号を求める際、剰余演算をするか否かを指定する単声イベント入力 (デフォルトはオフ)。オフであれば、シフト幅を

加算した結果が声部番号として無効な値になった場合、剰余演算を施します。例えば声部数が3のインストゥルメントでシフト幅を+1とした場合、声部3は声部1になります。

- ・ **Sh**: 声部番号をシフトする幅を制御する多声イベント入力。正值であれば番号を加算する方向、負値であれば減算する方向にシフトします。したがって、例えば **Sh**=1 であれば声部1は声部2に変わり、**Sh**=-2 であれば声部3は声部1に変わります。多声入力になっているので、声部ごとに別々のシフト幅を指定できます。何も指定しない場合のシフト幅は1です。
- ・ **In**: 変換元の多声信号入力。
- ・ **Out**: 変換処理を施された多声信号出力。

Audio Smoother (オーディオ・スムーサー)

補助系



単声イベント信号の値が滑らかに遷移するようにし、オーディオ信号として出力します。フェイダーやボタンの後段に置いて、値が連続的に変化するようにする使い方が典型的です。

入力イベント信号の急激な変化を、ある勾配の滑らかな変化に変換します。この勾配は、**Transition Time** プロパティ (ミリ秒単位) で調整できます。すなわち、入力値が急激に変化した場合、(その変動幅にかかわらず) 出力値はこの時間をかけて新しい値に遷移します。時間が長いほど平滑化の効果が高いことになります。

- ・ **In**: 変換元の単声イベント入力。
- ・ **Out**: 変換処理が施された単声オーディオ出力。

Event Smoother (イベント・スムーサー)

補助系



Audio Smoother と同様ですが、出力がイベント信号である点が異なります。

遷移中は、**Event Rate** プロパティで表される頻度でイベントを出力します。この頻度が高いほど、信号変化の分解能が高いことになります。

- ・ **In**: 変換元の単声イベント入力。
- ・ **Out**: 変換処理が施された単声イベント出力。

Master Tune/Level (マスター・チューン / レベル)

補助系



システム全体のレベルを調整し、調律を行います。

- ・ **Tun**: 調律用の制御入力。半音が単位となります。0.0 とすれば、A3 が 440Hz になります。通常の使用範囲: -1 ~ 1。
- ・ **Lvl**: 出力段でマスター・レベルを調整するための制御入力。1dB 単位で指定します。0.0 とすれば利得なし (0.0dB) になります。通常の使用範囲: -60 ~ 0。
- ・ **Tun**: 調律ピッチ設定値の出力。
- ・ **Lvl**: マスター・レベル設定値の出力。

Tempo Info (テンポ情報)

補助系

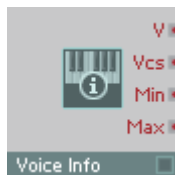


テンポの設定値を毎秒の拍数単位で返します。BPM 単位にしたい場合は 60 を掛けてください。

- ・ **Out**: テンポを毎秒の拍数単位 (Hz) で返すイベント出力。

Voice Info (声部情報)

補助系



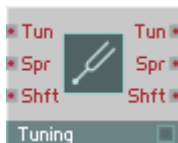
声部 (ボイス) に関する情報を返します。

- ・ **V**: 各声部の識別番号を返す多声出力。
- ・ **Vcs**: インストゥルメントに使われている声部数。
- ・ **Min**: インストゥルメントの **Min Unison Voices** の設定を出力。斉奏 (ユニゾン) で鳴らせる声部の最小数です。

- ・ **Max**: インストゥルメントの **Max Unison Voices** の設定を出力。
齊奏 (ユニゾン) で鳴らせる声部の最大数です。

Tuning Info (調律情報)

補助系

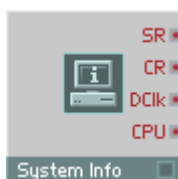


インストゥルメントの調律に関するパラメーターを設定し、また、現在の設定値を返します。

- ・ **Tun**: 半音を 1 とする単位でピッチを調整する制御入力。
- ・ **Spr**: 齊奏の場合に、各声部のピッチの幅を調整する制御入力。
- ・ **Shft**: 受け取った MIDI ノート・イベントの転調幅を調整する制御入力。
- ・ **Tun**: インストゥルメントのピッチ設定を返す出力。
- ・ **Spr**: 齊奏の各声部のピッチ幅設定を返す出力。
- ・ **Shft**: 転調幅の設定を返す出力。

System Info (システム情報)

補助系



システム情報、具体的にはサンプル・レート (毎秒のサンプル数)、コントロール・レート (Hz)、CPU に対する負荷 (%) を出力します。

- ・ **SR**: サンプル・レート (毎秒のサンプル数) の出力。
- ・ **CR**: コントロール・レート (Hz) の出力。
- ・ **DClk**: 画面表示レート (毎秒のフレーム数) の出力。画面を更新する直前に通知イベントを送信します。
- ・ **CPU**: CPU に対する負荷 (%) の出力。

Note Range Info (音域情報)

補助系



受信できる MIDI ノート・イベントの範囲 (音域) を設定し、また、現在の設定値を返すモジュールです。

- ・ **Upr**: 受信できる MIDI ノートの音域上限を指定する制御入力。
- ・ **Lwr**: 受信できる MIDI ノートの音域下限を指定する制御入力。
- ・ **Upr**: 受信できる MIDI ノートの音域上限設定値を出力。
- ・ **Lwr**: 受信できる MIDI ノートの音域下限設定値を出力。

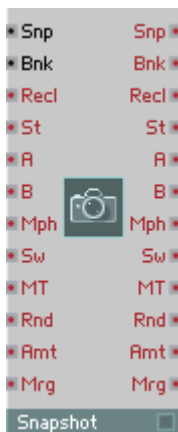
MIDI Channel Info (MIDI チャンネル情報)

補助系



MIDI チャンネルを設定し、また、現在の設定値を返すモジュールです。

- ・ **ICh**: 入力 MIDI チャンネルを選択する制御入力。
- ・ **OCh**: 出力 MIDI チャンネルを選択する制御入力。
- ・ **ICh**: 入力 MIDI チャンネルの設定値を出力。
- ・ **OCh**: 出力 MIDI チャンネルの設定値を出力。



スナップショットを切り替えたり、2つのスナップショット間を補間 (モーフィング) したりするためのモジュールです。List モジュール (パネル系モジュールの節を参照) と同等の機能が組み込まれています。

Appearance ページの設定

このモジュールのパネル表示形式は、**Properties** ウィンドウの **Appearance** ページで、次の中から選択できます。

- **Button:** モジュールの入力ポートごとに、ボタンを縦に並べて表示します。オンになっているボタンは、インストゥルメントのインジケータ色になります。
- **Menu:** 入力ポート名を項目として列挙した、ドロップ・ダウン・メニューの形で表示します。
- **Text Panel:** 入力ポート名を項目として列挙し、全体が一度に見えるようにしたリストとして表示します。**Size X** および **Size Y** で表示の大きさを指定しますが、そこに表示しきれない場合はスクロール・バーが現れます。
- **Spin:** 入力ポート名を項目として列挙したリストとして表示します。各項目の右側には + ボタン、- ボタンがあって、値を切り替えることができます。

Size X、**Size Y** 欄には、パネルに表示する際の大きさを指定します。

ポート

- ・ **Snp**: 呼び出す、または保存するスナップショットの番号を選ぶオーディオ入力。範囲: 1 ~ 128。
- ・ **Bnk**: スナップショット・バンクを選ぶオーディオ入力。範囲: 1 ~ 16。
- ・ **Recl**: 正の値を与えれば、**Snp** および **Bnk** で指定されたスナップショットを呼び出します。
- ・ **St**: 正の値を与えれば、**Snp** および **Bnk** の指定に従ってスナップショットを保存します。
- ・ **A**: 正の値を与えれば、**Snp** および **Bnk** で指定されたスナップショットを、モーフィングする一方 (**A** 側) として呼び出します。
- ・ **B**: 正の値を与えれば、**Snp** および **Bnk** で指定されたスナップショットを、モーフィングするもう一方 (**B** 側) として呼び出します。
- ・ **Mrph**: **A** および **B** に呼び出したスナップショットを混ぜ合わせる (モーフィング) 比率を指定します。0.0 以下ならば **A** のみ、0.0 ~ 1.0 ならば **A** と **B** の間を補間、1.0 以上ならば **B** のみになります。
- ・ **Sw**: 負の値または 0 を与えれば、スナップショット **A** 側に切り替わります。正の値を与えれば **B** 側に切り替わります。
- ・ **MT**: モーフィングを行う時間をミリ秒単位で指定するイベント入力。
- ・ **Rnd**: 正の値を与えれば、スナップショットのランダム化機能が起動されます。
- ・ **Amt**: ランダム化の幅を指定する入力。範囲: 0.0 ~ 1.0、(1.0 = 100%)。
- ・ **Mrg**: 正の値を与えれば、ランダム・マージ機能が起動されます。
- ・ **Snp**: 現在呼び出しているスナップショットの番号を出力。スナップショットを呼び出す、あるいは保存するたびにイベントを送信します。
- ・ **Bnk**: 現在呼び出しているスナップショットのバンク番号を出力。スナップショットを呼び出す、あるいは保存するたびにイベントを送信します。
- ・ **Recl**: スナップショットを呼び出したあと、値が 1.0 のイベントを送信します。
- ・ **St**: スナップショットを保存したあと、値が 1.0 のイベントを送信します。

- ・ **A:** モーフィングの **A** 側に呼び出したスナップショットの番号を出力。
A 側にスナップショットを呼び出す、あるいは保存するたびにイベントを送信します。
- ・ **B:** モーフィングの **B** 側に呼び出したスナップショットの番号を出力。
B 側にスナップショットを呼び出す、あるいは保存するたびにイベントを送信します。
- ・ **Mrph:** **A** および **B** に呼び出したスナップショットを混ぜ合わせる (モーフィング) 比率を出力。0.0 以下ならば **A** のみ、0.0 ~ 1.0 ならば **A** と **B** の間を補間、1.0 以上ならば **B** のみになります。
- ・ **Sw:** スイッチ / ボタンの状態を出力。スナップショット **A** 側にあれば 0.0、**B** 側にあれば 1.0 を返します。
- ・ **MT:** モーフィングを行う時間 (ミリ秒単位) の出力。
- ・ **Rnd:** ランダム化機能を起動したあと、値が 1.0 のイベントを送信します。
- ・ **Amt:** ランダム化の幅を出力。範囲: 0.0 ~ 1.0、(1.0 = 100%)。
- ・ **Mrg:** ランダム・マージ機能を起動したあと、値が 1.0 のイベントを送信します。

Set Random (乱数の種)

補助系



Random、**Slow Random**、**Geiger** の各モジュールで使う擬似乱数生成器の種を設定します。同じインストゥルメントの中のモジュールのみが影響を受けます。種を変更すれば、異なる系列の乱数が生成されるようになります。

Unison Spread (ユニゾン・スプレッド)

補助系



斉奏 (ユニゾン) で再生する各声部のパラメーターをわずかに変化させるための固定値を出力する多声イベント源。

斉奏モードで同じ音を再生する各声部は、いくつかのパラメーターをわずかに変えることにより、単に 1 つの声部を再生した場合とは異なる、厚みのある響きを実現できます。ピッチに関してはインストゥルメントの **Unison Spread** パラメーターに従って自動調整するようになってい

ますが、それ以外のパラメーターについては、このモジュールの出力値を加算することにより効果を得てください。

入力値はパラメーターの変動幅を表します。その範囲内で、声部ごとに少しずつ異なる値を出力します。値は次の音符に切り替わる時点で変わります。

Snap Value (スナップショットの付値)

補助系

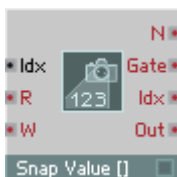


スナップショットと組にして、ある値を保存します。あとでスナップショットを呼び出せば、この値が出力されるようになります。

- ・ **In:** スナップショットと組にして保存する値の指定。保存する時点でこのポートの値を調べます。イベント源につながっていれば、入力イベントはそのまま出力されます。
- ・ **Out:** 呼び出したスナップショットに保存されていた値を出力します。

Snap Value Array (スナップショットの付値 - 配列)

補助系



一連の浮動小数点数の配列を、スナップショットと組にして保存し、必要に応じて再び読み込みます。

このモジュール 1 つで最大 40 組の配列を保持できます。各配列の要素数には、システム・メモリー量の制約以外に制限がありません。但し各配列の要素数は同じであるとしします。配列の個数、各配列の要素数は、プロパティーとして設定します。

メモリー領域は動的に確保します。すなわち、スナップショットを作ること必要量を確保し、不要になれば解放します。したがって必要最小限のメモリーしか消費しません。

Snap Isolate プロパティーをオンにすると、直近に各要素に与えた値のみを保存するようになります (アンサンブルの初期設定時にはこの

値を復元)。スナップショットに対する読み書き操作により、配列のデータが影響を受けることはありません。

典型的な用途としては、シーケンス・データを配列の形で保存する、というものがあります。この場合、**Event Table**、**Multi Display**などのモジュールと組み合わせて使います。

このモジュールの入力は、いずれも単声信号しか受け付けません。

- ・ **Idx**: 配列上のある要素を特定するためのインデックス (読み書き両用)。先頭要素を 1 として数えます。小数が与えられた場合は四捨五入して扱います。範囲外の値が与えられた場合の動作は **Index Behaviour** プロパティで設定できます。
- ・ **R**: イベントを与えるとそれに同期して、**Idx** で指定される要素の値が **Out** から出力されます。言い方を変えると、**R** 入力に与えたイベントは、インデックスで示される値に変わり、**Out** を経由して次のモジュールに伝播する、ということです。配列が複数あれば、それぞれについて同じインデックスの要素が並列に伝播します。なお、ここにイベントを与える前に **Idx** 値を設定しておかなければなりません。
- ・ **W**: イベントを与えるとそれに同期して、**Idx** で指定される要素の値がイベントの値で上書きされます。配列が複数あれば、それに応じた個数の **W** ポートができます (配列の数はプロパティとして指定)。必要に応じ、各 **W** ポート名を変えても構いません。**Events Thru** プロパティをオンにすると、**W** で受け取ったイベントを対応する **Out** からそのまま出力するようになります。
- ・ **N**: 各配列の要素数を出力。
- ・ **Gate**: **Out** からイベントを送出する前にゲートを開くイベント (値 1)、送出した後でゲートを閉じるイベント (値 0) を出力します。
- ・ **Idx**: 現在読み書きしている要素のインデックスを出力。**R** や **W** でイベントを受け取ったときのほか、スナップショット操作 (呼び出し、モーフィングなど) の際にも出力します。読み込み操作の場合は、先にこの **Idx** にインデックスを出力した後で **Out** から値を送出します。
- ・ **Out**: 読み込み操作 (**R** にイベントを入力) やスナップショット操作 (呼び出し、モーフィングなど) に応じ、**Idx** で指定された要素の値を出力。**Self-Iteration** プロパティをオンにすると、初期設定、アクティブ化、スナップショットの呼び出し、ランダム化、ランダム・マージ、モーフィングの各操作のとき、配列の各要素を先頭から (インデックス 1 番から) 順に出力するようになります。また、スナップショットを呼び出すことにより、配列の各要素を一斉に更新できます。

入出力端子系モジュール

オーディオ信号や制御信号を、インストールメントやマクロのストラクチャー外部とやり取りするための端子 (ターミナル) モジュールです。ストラクチャー・ウィンドウ内で、**Ctrl** キーを押しながらインストールメントやマクロ上にケーブルをドラッグすると、自動的に入出力端子系モジュールができます。

In Port (入力ポート)

入出力端子系



外部からインストールメントやマクロにオーディオ信号や制御信号を印加するための入力端子。

Out Port (出力ポート)

入出力端子系



インストールメントやマクロから外部にオーディオ信号や制御信号を送出するための出力端子。

Send (信号送信)

入出力端子系



インストールメント内で、ケーブルなしの接続を通してオーディオ信号や制御信号を送信するための端子。

Send モジュールを挿入すると、**Receive** モジュールのプロパティ・ウィンドウに、送信元の選択肢として追加されます。

Receive (信号受信)

入出力端子系



インストールメント内で、ケーブルなしの接続を通してオーディオ信号や制御信号を受信するための端子。

プロパティー - Function ページ

インストゥルメントに **Send** モジュールを追加すると、それに応じて自動的にリストに項目が追加されます。項目名は **Send** モジュールのラベルをもとにして決まります。リストの上にある **Up** ボタン、**Down** ボタンで、項目を上下に動かして順序を変えることができます。

リストには次のような欄があります。それぞれの見出しをクリックすれば、その欄を基準として並べ替えられます。

- ・ **#:** **Send** モジュールの、リスト上の順序番号。 **Default** 欄には、この番号で指定します。
- ・ **Label:** **Send** モジュールの名前。モジュール名を変更すればこの欄も自動的に更新されます。
- ・ **State:** このモジュールに接続できる状態か否かを示します。 **OK** と表示されている場合に限って接続が可能です。
- ・ **Use:** この欄をクリックすると、あらかじめ選択しておいた **Send** モジュールとの間を接続できます。既に他のモジュールと接続されている場合は「x」と表示されます。
- ・ **Mouse Resolution:** マウスの動きの分解能を表します。表示スタイルが **Spin** の場合、マウスでクリックして上下に動かすことにより値を増減できますが、その感度を調整します。

このモジュールを初期化すると、**#** 欄のうち、**Default** 欄に指定された番号に一致する行が選択状態になります。

プロパティー - Appearance ページ

このモジュールのパネル表示形式は、**Properties** ウィンドウの **Appearance** ページで、次の中から選択できます。

- ・ **Button:** モジュールの入力ポートごとに、ボタンを縦に並べて表示します。オンになっているボタンは、インストゥルメントのインジケータ色になります。
- ・ **Menu:** 入力ポート名を項目として列挙した、ドロップ・ダウン・メニューの形で表示します。
- ・ **Text Panel:** 入力ポート名を項目として列挙し、全体が一度に見えるようにしたリストとして表示します。 **Size X** および **Size Y** で表示の大きさを指定しますが、そこに表示しきれない場合はスクロール・バーが現れます。

- ・ **Spin**: 入力ポート名を項目として列挙したリストとして表示します。各項目の右側には + ボタン、- ボタンがあって、値を切り替えることができるようになっていきます。

Size X、**Size Y** 欄には、パネルに表示する際の大きさを指定します。

IC Send (アンサンブル内送信)

入出力端子系



IC(Internal Connection) 通信に対応したモジュールに、単声イベント信号を送信するモジュール。代表的な送信先として **IC Receive** モジュールがありますが、ノブやスイッチなど、制御パネル上のつまみ類に対しても送信できます。アンサンブル内であれば、別のインストールメントとの間でも、結線なしで通信可能です。

IC Send モジュールにも制御パネルがあって、インストールメントの制御パネルから呼び出す形で使えます。アンサンブル内にある、通信可能なモジュールがここに列挙されています。但し **No Entry in IC Menu** プロパティがオンになっているものを除きます。この制御パネル上でも通信相手を設定可能です。

IC Receive (アンサンブル内受信)

入出力端子系



IC(Internal Connection) 通信プロトコルに従って送られてきた単声イベント信号を受け取り、出力するモジュール。一般には **IC Send** モジュールと対にして使いますが、ノブやスイッチなど、制御パネル上のつまみ類とも通信できます。

プロパティとして通信相手を設定できるほか、**IC Send** モジュールとの接続に関しては、**IC Send** モジュール側の制御パネルでも設定可能です。

OSC Send (OSC 送信)

入出力端子系



OSC メッセージのやり取りは、**OSC Send** および **OSC Receive** の 2 種類のモジュールで行います。どちらもポート数は動的に増やせるようになっています。**Ctrl** キーを押しながらワイヤーをつかみ、ポート領域 (**OSC Send** は左端、**OSC Receive** は右端) の空いているところに持って行ってください。

OSC Send モジュールに複数の接続を施すと、OSC メッセージに複数のパラメーターが並ぶようになります。例えば XY モジュールの出力 **MX** および **MY** を **OSC Send** モジュールに入力するよう接続すれば、1 つの OSC メッセージに X および Y の 2 つのパラメーターが添えられます。なお、**OSC Receive** モジュールでは複数のパラメーターが並んだ OSC メッセージを受信できますが、その最大数は 10 となっています。

OSC Send モジュールに複数の入力ポートを設けた場合でも、OSC メッセージのトリガーをかけるのは常に 1 番目のポートになります。

OSC メッセージを直接送受信できるモジュールも、パネル系などにいくつかあります。この場合も送受信に関する設定方法は同じです。

入力ポート数は動的に変更できます。**Function** ページの **Min Num Port Groups** プロパティとして指定してください。

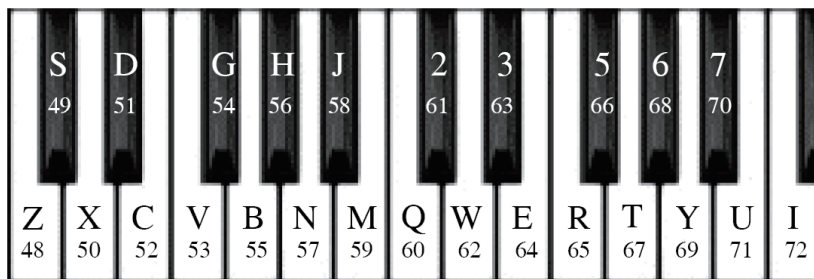
OSC Receive (OSC 受信)

入出力端子系



OSC Send モジュールの項を参照してください。

付録 音符に対応するコンピューター・キーボード上のキー



- ・ **Shift** を押しながら操作すると 2 オクターブ高い音になります (+24)。
- ・ Windows ならば **Ctrl+Shift**、MacOS X ならば **apple+shift** を押しながら操作すると、2 オクターブ低い音になります (-24)。

索引

数字

0-1 to 0-127 Range Converter (マクロ) 263

0...1 232, 233

12-Step (モジュール) 350

16-Step (モジュール) 350

16 ステップ・シーケンサー + ベースライン 61

1/96 Clock (モジュール) 288, 293

1 / Square Root (モジュール) 302

2D Color 235

2D Curve 235

2D Table Default 140

2D Table Max 140

2D Table Min 140

3 Band Filter (マクロ) 258

3D Frame 152

4-Ramp (モジュール) 326, 365

4-Step (モジュール) 325

5-Ramp (モジュール) 326, 366

5-Step (モジュール) 325

6-Ramp (モジュール) 326, 366

6-Step (モジュール) 325, 349

8-Ramp (モジュール) 326

8-Step (モジュール) 325, 350

アルファベット

A

A 131, 141, 152

$a * b + c$ (モジュール) 297

AB 141, 152

About 105

Accumulator (モジュール) 398

Activate MIDI In 185, 194

AD-Env (モジュール) 358

ADBDR-Env (モジュール) 361

ADBDSR-Env (モジュール) 362

Add 157, 221

Add (モジュール) 295

ADR-Env 68

ADR-Env (モジュール) 359

ADSR 78

ADSR-Env (モジュール) 360

AHDBDR-Env (モジュール) 364

AHDSR-Env (モジュール) 363

Akai Import 221

ALIGNMENT 235, 236

ALL 143

ALL CONTROLS 141

Allpass 1-Pole (モジュール) 368

Amp - Exponential (マクロ) 245

Amp - Linear (マクロ) 245

Amp/Mixer 73, 157

Amp (モジュール) 308

AND 401

ANIMATION HEIGHT 198

ANIMATION WIDTH 198

APPEND 179, 205

Apply 49

Apply to all 226

AR-Env (モジュール) 358

Arccos (モジュール) 303

Arcsin (モジュール) 303

Arctan (モジュール) 304

Arrange Icons 105

ASIO 25
A to E (Perm) (モジュール) 414
A to E (Trig) (モジュール) 413
A to E (モジュール) 413
A to Gate (モジュール) 414
AU 26
Audio + MIDI Settings 27
Audio + MIDI Settings... 94, 120
AUDIO FILES 99
AUDIO IN 108
AUDIO OUT 109
Audio Setup 27
Audio Smoother (モジュール) 416
Audio Table 229
Audio Table (モジュール) 329
Audio Units 26, 40
Audio Voice Combiner (モジュール)
412
Automatic Layout 91
Automatic Panel Layout 126
Automatic saving by default in this
song 34
Automatic Voice Reduction 136
AUTOMATION 187
AUTO SAVE FOLDER 35
Available in Panelsets 140

B

B 131, 141, 152
BACKGROUND BITMAP 152
BACKGROUND PICTURE 141
Backup Data with Module 231
Bandsplit (マクロ) 258
Bar 235
Batch Processing 88

Beat Loop (マクロ) 249
Beat Loop (モジュール) 346
Bi-Pulse (モジュール) 322
Bi-Saw (モジュール) 311
Bookmark 110
BORDER BOTTOM 200
BORDER LEFT 200
BORDER RIGHT 200
BORDER TOP 200
BPM 110, 232
Browser 109, 110
Browser and Snapshots always on top
95
Built-In Module 170
Button (モジュール) 269

C

Cascade 105
Ch. Aftertouch (モジュール) 286, 292
CHANNEL 127, 143, 144
Channel Message 4
Channel Message (モジュール) 289,
294
Chopper (モジュール) 392
Classic Modular 241
Classic Sampler (マクロ) 249
CLIENTS 231
Clip 231
Clipper (マクロ) 261
Clipper (モジュール) 391
CLIP/WRAP XY 231
Clock Oscillator (モジュール) 327
Clock Start 91
Clock Stop 91
CLOCK SYNC 49

Clone 208
Close 105
Close All Structures 105
COLOR SCHEME 126, 140
Comb (マクロ) 258
COMPARE 205
Compare/Equal (モジュール) 299
Compare (モジュール) 299
Compress 33
COMPUTER KEYBOARD MIDI
 CHANNEL 96
CONNECTIONS 144, 187
Constant 161
Constant (モジュール) 295
CONT/NOTE 187, 194
CONTROLLER 128
Controller 187, 194
Controller (マクロ) 243
Controller (モジュール) 286, 291
CONTROL RATE 125
Control Rate 90
Copy 89, 131, 147, 158, 239
Core 2
Core Audio 25
Core Cell 170
Counter (モジュール) 398
CPU 108
Create backup file before saving 95
Create Constant 158
Create Control 157
Crossfade (マクロ) 245
Crossfade (モジュール) 306
Ctrl. Shaper 1 BP (モジュール) 400
Ctrl. Shaper 2 BP (モジュール) 400

Ctrl. Shaper 3 BP (モジュール) 401
Cubase 37
Cut 89, 131, 147, 158, 239

D

D-Env (モジュール) 354
Data 223
DBDR-Env (モジュール) 356
DBDSR-Env (モジュール) 357
Debug 92, 111
DEFAULT 205
Delay (マクロ) 260
DELETE 179, 205
Delete 89, 131, 148, 158, 208, 221
Delete Rows 239
DEVICE 143, 144
Differentiator (モジュール) 382
Diffuser Delay (モジュール) 384
Digital Performer 41
DIRECTORY 102
DirectSound 25
Disable 187
DISPLAY UNITS 232
Distributor (モジュール) 306
Divide (モジュール) 297
DR-Env (モジュール) 355
DSR-Env (モジュール) 355
Duplicate 89, 131, 148, 158
DXi2 26
DXi 2 26, 44

E

Edit 214, 221
EDIT COLORS 126, 139
Edit Sample List 220

ENABLE GRID 236
ENSEMBLE NAME 35
Ensemble Panel 110
Ensemble Structure 109
ENTRIES 179
Envelope - ADSR (マクロ) 256
Envelope - Decay (マクロ) 256
Envelope - One-Ramp (マクロ) 256
Envelope Follower (マクロ) 256
EVENT LOOPS 124, 138, 149
Event Smoother (モジュール) 416
Event Table 229
Event Table (モジュール) 407
Event V.C. - All (モジュール) 412
Event V.C. - Max (モジュール) 412
Event V.C. - Min (モジュール) 413
Exclusive OR 402
Exit 88
Expon. (A) (モジュール) 300
Expon. (F) (モジュール) 300
EXT 132
EXTERNAL SAMPLE EDITOR 99
External Sync 91

F

FACTORY CONTENT 98
Fader (モジュール) 266
FILE 100, 103, 231
File 223
FM Overdrive 59
FRAME STYLE 152
Frame with Label 152
Frequency Divider (モジュール) 397,
399
From Voice (モジュール) 415

FUNCTIONS 220

G

GAIN (DB) 220
Garage Band 43
Gate 68, 78, 190
Gate (モジュール) 285
Geiger Counter (マクロ) 247
Geiger (モジュール) 329
Global Clock (マクロ) 250
Globally disable event loops 96
Grain Cloud Delay (モジュール) 387
Grain Cloud (モジュール) 344
Grain Delay (モジュール) 385
Grain Pitch Former (モジュール) 340
Grain Resynth (モジュール) 336
GRAPH 235
Graph BG 140
Graph Fill 140
Graph Line 140
GRID 236
Grid 140

H

H-Env (モジュール) 353
HEAD (PIXELS) 175, 178
Hide Groove 175
Hide Scale 175
High Shelf EQ FM (モジュール) 380
High Shelf EQ (モジュール) 379
HOLD CTRL 143
Hold (モジュール) 406
Horizontal 200
Horizontal 198
HP/LP 1-Pole FM (モジュール) 368

HP/LP 1-Pole (モジュール) 367
HR-Env (モジュール) 354
H Scroll Bar 235
HVel 222

I
IC Receive 4
IC Receive (モジュール) 427
IC Send 4
IC Send (モジュール) 427
ID 188
Ignore Root Key when Loading 224
Ignore Tempo Change 92
IMPORTED FILES (AKAI) 99
Import MIDI File 115
Import MIDI File... 87
Impulse FM (モジュール) 323
Impulse Sync (モジュール) 324
Impulse (モジュール) 323
In 132
Includes Frame 152
Incremental 186, 195
Index 232, 233
Indicator 140
INFO 125, 138, 150
Init 208
In Port (モジュール) 425
INSERT 179, 205
Insert Rows 239
Inst 1 40
Instrument 170
Instrument down 33
Instrument/Macro Properties 170
Instrument up 33
INT 132

Integrator (モジュール) 382
Interface 28
INTERNAL SAMPLE RATE 124
INTERPOLATION 230
Invert 4
Inverter (マクロ) 245
Invert (モジュール) 296
ITEM 140
Iteration (モジュール) 403

J

Jump to Bookmark 111

K

Knob (モジュール) 268

L

L 215, 222
LABEL 123, 134, 149
Label 236
Ladder Filter FM (モジュール) 377
Ladder Filter (モジュール) 377
Ladder Lowpass (マクロ) 259
Lamp 4
Lamp (モジュール) 272
LENGTH 100, 103, 175, 220
Level Lamp (モジュール) 273
LevelMeter (モジュール) 275
LFO 352
LFO 系 256, 352
LFO (マクロ) 257
LFO (モジュール) 352
Line 235
Line Frame 152
LINKED 204

List (モジュール) 270
Load 100, 102, 208, 231
Load Data into Table 237
Load Map 221
LOCAL IDENTIFIER 49
LOCAL IP ADDRESS 49
LOCAL PORT 49
Location 223
Lock/Unlock Panel 132
Lock Voices 135
Log (A) (モジュール) 301
Log (F) (モジュール) 301
Logic 40
Logic AND 4
Logic AND (モジュール) 401
Logic EXOR (モジュール) 402
Logic NOT 4
Logic NOT (モジュール) 402
Logic OR 4
Logic OR (モジュール) 401
Logic XOR 4
Loop 100, 103
LOOP LENGTH (BARS) 102
Loop MIDI File 92
Loop On 226
LOWER NOTE 143
Low Shelf EQ FM (モジュール) 381
Low Shelf EQ (モジュール) 380
LVel 222

M

Macro 170
MAP NAME 220
Map View 220
Master 49, 50

MASTER TUNE/LEVEL 124
Master Tune/Level (モジュール) 417
Master Volume (マクロ) 245
MAXIMUM CPU USAGE 97
MAX UNISON V 135
Measure CPU Usage 92
Merge 4, 157
Merge (モジュール) 404
Meter 4
Meter (モジュール) 274
MIDI Channel Info (モジュール) 419
MIDI Clock Out 91
MIDI IN 109, 142
MIDI In 159
MIDI In Controller 4
MIDI IN DEVICE 127
MIDI Learn 90, 110, 180
MIDI Note 187, 194, 232
MIDI OUT 109, 144
MIDI Out Controller 4
Midi Track 42
MIDI コントローラー (マクロ) 243
MIDI 出力系 291
MIDI 処理系 243
MIDI 信号源 159
MIDI タブ 31
MIDI チャンネル情報 419
MIDI 入力系 284
Milliseconds 232
Minimize 105, 132
MIN UNISON V 135
Mirror 1 Level (モジュール) 392
Mirror 2 Levels (モジュール) 392
Mirror X 239

Mirror Y 239
Mixer - Simple (マクロ) 245
Mixer - Studio (マクロ) 246
Mixer (モジュール) 308
MME 25
Mod. Clipper (モジュール) 391
Modulation Matrix - Mixer (マクロ)
246
Modulation Matrix - Switch (マクロ)
246
Modulation Mixer (マクロ) 246
Modulo (モジュール) 298
MONITOR OPTIONS 51
Mono 147, 149, 158
MORPH 127
MORPH CTRL 143
Mouse Area 3
Mouse Area (モジュール) 281
Multi-Ramp (モジュール) 326
Multi-Sine (モジュール) 318
Multi-Step (モジュール) 325
Multi-Tap Delay (モジュール) 384
Multi 2-Pole FM 79
Multi 2-Pole FM (モジュール) 369
Multi 2-Pole (モジュール) 369
Multi Display 3
Multi Display (モジュール) 279
Multi/HP 4-Pole FM (モジュール) 375
Multi/HP 4-Pole (モジュール) 374
Multi/LP 4-Pole FM (モジュール) 373
Multi/LP 4-Pole (モジュール) 372
Multimode - Accurate (マクロ) 259
Multimode - Resonance Limiter (マク
ロ) 260

Multi/Notch 2-Pole FM (モジュール)
371
Multi/Notch 2-Pole (モジュール) 370
Multi Picture 4
Multi Picture (モジュール) 275
Multiplex 16 (モジュール) 351
Multiply 81
Multiply (モジュール) 296
Multi Text 4
Multi Text (モジュール) 276
Mute 120, 130, 135, 147, 149, 158
Mute audio during Core compilation
96
Mute Port 158

N
NAME 197
Name 222
Native Instruments 19
Nearest 136
New 102, 231
New-2In2Out 65, 78
New Bank 208
New Ensemble 64, 77, 86
Newest 136
Next Y 238
No Frame 152
Noise (マクロ) 247
Noise (モジュール) 328
NONE 143
None 230
NOT 402
Note Pitch 68, 78
Note Pitch/Gate (モジュール) 291
Note Pitch (モジュール) 284

Note Range Info (モジュール) 419
Notes - Monophonic (マクロ) 243
Notes - Polyphonic (マクロ) 243
NOTE SHIFT 143
Nuendo 38
NUM ANIMATIONS 198
Number Display (マクロ) 242
NUMBER OF UNDOS 96
NUM ENTRIES 179
Numeric 232
NUM. OF BEATS 232

O

Off Velocity (モジュール) 285
Oldest 136
Only if changed 137
On Velocity (モジュール) 285
Open 108
Open... 53, 86
Open Sound Control 47
Options 220
OR 401
Order 4
Order (モジュール) 402
OSC 47, 144
OSC CONNECTIONS 128
Oscillator - Symmetry (マクロ) 247
Oscillator - Sync (マクロ) 248
OSC MESSAGE 51
OSC MONITOR 51
OSC Receive (モジュール) 428
OSC Send (モジュール) 427
OSC Settings 48, 187
OSC Settings... 94
OSC SOURCE 128, 144, 187

OSC TARGET 128, 187
OSC 監視 51
OSC クライアント 50
OSC 識別名 49
OSC 同期処理 49
OSC メンバー・リスト 50
Out 132
Out Port (モジュール) 425
Output Device 28
Output Latency 28
OVERWRITE 205
OWNER 220

P

Pad Echo 55
page down 193
page up 193
PAN 220
Panel 140
PANEL COLOR 126
Panel Index 3
Panel to MIDI 186, 195
Panner (マクロ) 246
Panner (モジュール) 306
Parabol (モジュール) 314
Parent 170
Parent Window 170
Par FM (モジュール) 314
Par PWM (モジュール) 315
Par Sync (モジュール) 315
Paste 89, 170, 239
Pause 100, 103
Pause/Stop Clock 109
Peak Detector (モジュール) 396
Peak EQ FM (モジュール) 379

Peak EQ (モジュール) 378
Picture 234
PICTURE INDEX 140, 141, 151, 152
Picture (モジュール) 275
Pitch + Gate 75
Pitchbend (モジュール) 284, 291
Pixel 235
Play 100, 103
PLAYED 100, 103
Play MIDI File 92
Poly Aftertouch 187, 194
Poly Aftertouch (モジュール) 287, 292
Poly Display 3
Poly Display (モジュール) 279
Position Delay (マクロ) 250
Position Looper (マクロ) 250
Position Offset (マクロ) 251
Power (モジュール) 301
Preferences 94
PREVIEW 200
Previous Y 238
Pro-52 Filter (モジュール) 376
Program Change (モジュール) 287, 293
Properties 109, 110, 120, 131, 148, 158, 180
Pro Tools 45
Pulse 1-ramp (モジュール) 321
Pulse 2-ramp (モジュール) 321
Pulse FM (モジュール) 319
Pulse Sync (モジュール) 320
Pulse (モジュール) 318

Q

Quantizer (マクロ) 261, 263
Quantize Value to Step Size 239
Quantize (モジュール) 300

R

R 215, 222
Ramp (モジュール) 327
RAND. AMOUNT 209
RAND. MERGE 209
RANDOMIZE 208
Randomizer (マクロ) 263
Randomizer (モジュール) 399
Random (マクロ) 248
Random (モジュール) 328
REACTOR Core 2
REAKTOR 4 LEGACY MODE 124
REAKTOR 5 Registration Tool 10, 12, 15
Reassign 136
Recall by MIDI 124, 137
Recall by Parent 137
Recall Panelset 104
RECEIVE MIDI 185
Receive (モジュール) 425
Reciprocal (モジュール) 297
Record 102
RECORD START BY 102
RECORD STOP BY 102
Rectifier (モジュール) 298
Rectify 4
Rectify/Sign (モジュール) 298
Redo 89
Registration Tool 8

Rel 226
Relay (モジュール) 306
Reload 214
Reload last ensemble at startup 95
Reload Table Data 237
Remap to Single Keys 224
Remote to MIDI 187, 195
Replace 221
Reset All Tool Window Positions 104
RESIZABILITY 200
Resynth (マクロ) 249
RGB Lamp (モジュール) 273
RGB ランプ 273
Ringmodulator (マクロ) 262
Root 215, 222
Rotate/Add/Scale... 239
Routing タブ 29
RTAS 26
Run/Stop Audio 92, 108

S

Sample and Hold (マクロ) 258
Sample Lookup (モジュール) 348
SAMPLE RATE 108, 220
Sample Rate 90
Sample rate 28
Sampler FM (モジュール) 333
Sampler Loop (モジュール) 334
Sampler (モジュール) 332
SAMPLES/SEC 232
SAMPLES/TICK 232
Saturator 2 (モジュール) 390
Saturator (マクロ) 262
Saturator (モジュール) 390
Save 108, 208, 221, 231

Save Ensemble 86
Save Ensemble... 77
Save Ensemble As... 86
Save Instrument As... 131
Save Instrument/Macro As... 170
Save Macro As... 148
Save Map 221, 228
Save Table Data 237
Save Table Data as... 237
Save this instance automatically 34
Save Window As... 86
Saw FM (モジュール) 309
Saw Pulse (モジュール) 311
Saw Sync (モジュール) 310
Sawtooth (モジュール) 309
Scanner (マクロ) 247
Scanner (モジュール) 305
Scope (モジュール) 278
SEL BY KEY 220
Select All 89, 170, 238
SELECT INSTRUMENT 204
Selective Gates - MIDI Keyboard (マクロ) 243
Selective Gates - QWERTY - Lower
Keys (マクロ) 244
Selective Gates - QWERTY - Upper
Keys (マクロ) 244
SELECT MASTER 50
Selector (モジュール) 305
Select X All 238
Select Y All 238
Sel. Note Gate (モジュール) 285
Sel. Poly AT (モジュール) 287, 292
Send (モジュール) 425

Separator (モジュール) 403	Show/Hide Playerbox 99
Sequencer - 1x Notes, 4x Mod, 8x Trigger (マクロ) 251	Show/Hide Properties 103
Sequencer - Classic Step (マクロ) 251	Show/Hide Recorderbox 101
Sequencer - Modulation 4x (マクロ) 252	Show/Hide Sample Map Editor 103
Sequencer - Note (マクロ) 253	Show/Hide Snapshots 103
Sequencer - Simple Modulation (マク ロ) 254	Show/Hide Toolbox 99
Sequencer - Trigger 8x (マクロ) 255	Show Horizontal Position Line 238
Sequencer (モジュール) 349	Show Horizontal Scroll Bar 238
Set 231	Show in Structure 180
Set 2D Draw Value... 239	Show Map Editor 217
Set BPM by Snapshot 124	Show Module Sorting 93, 169
Set Protected 90	Show Panel 85
Set Random (モジュール) 422	Show Properties 122, 133, 148
Set to Default 180	Show Read Position 238
Set Transpose To Null For All 224	Show Sample Names 224
Set Unprotected 90	Show Selection 237
Shaper 1 BP (モジュール) 393	Show Snapshots 203, 204
Shaper 2 BP (モジュール) 393	Show Vertical Scroll Bar 238
Shaper 3 BP (モジュール) 394	Show Write Position 238
Shaper Cubic (モジュール) 395	Simple Scope (マクロ) 242
Shaper Parabolic (モジュール) 394	Sine/Cos (モジュール) 303
Show All 237	Sine FM (モジュール) 316
Show Browser 113	Sine Sync (モジュール) 317
Show Event Initialization Order 93	Sine (モジュール) 302, 316
Show/Hide Browser 103	Single Delay (モジュール) 383
Show/Hide Hints 99	Single Key Mode 224
Show/Hide Info 110, 111	Single Trig. Gate (モジュール) 285
Show/Hide Loop Editor 220	SIZE 174, 176, 178
Show/Hide Panel 104	SIZE X 178, 235
Show/Hide Panelset Bar 109	SIZE Y 178, 235
	SKIN BITMAP 175, 176, 178
	Slave Player Controls to Recorder 102
	Slew Limiter (マクロ) 262

Slew Limiter (モジュール) 395
Slow Random (モジュール) 353
Small Label/Value 236
Snap Selection to Grid 239
SNAPSHOT 124
Snapshot 132
Snapshot Master for PlugIn 124, 138
Snapshots 109
Snapshot (モジュール) 420
Snap Value 4
Snap Value Array 4
Snap Value Array (モジュール) 423
Snap Value (モジュール) 423
Soft Takeover 185, 195
Solid 235
Solo 130, 135
Sonar 44
Song Pos (モジュール) 289, 293
Sort 33, 208
Soundcard タブ 27
Square Root (モジュール) 302
Stacked Macro 3
Stacked Macro (モジュール) 283
START OFFSET (BARS) 102
Start/Restart Clock 109
Start/Stop (モジュール) 288, 293
STATUS 123, 134, 149
Step 236
Step Filter (モジュール) 404
Stereo Amp (モジュール) 308
Stereo Mixer (モジュール) 308
Stereo Pan (モジュール) 307
Stop 100, 103
Store by Parent 137

Store Map with Module 212
Store Panelset 104
Structure 131, 148
STRUCTURE ICON 140, 151
Structure Window 131, 148
STYLE 178
Subtract (モジュール) 296
SUSTAIN CTRL 143
SWITCH 178
Switch (モジュール) 271
Sync Clock (モジュール) 288
SYNC MESSAGES 50
SynthOne 53
System ID 8
System Info (モジュール) 418
Sz1 236
Sz2 236
Sz3 236
Sz4 236

T

Table Control Mode 237
Table Draw Mode 237
TABLE FILES 99
Table Select Mode 237
Tapedeck 1-Ch (モジュール) 409
Tapedeck 2-Ch (モジュール) 411
TARGET 144
Tempo Info (モジュール) 417
Tempo Ticks 232
Terminal... 162
Text (モジュール) 276
TICKS/BEAT 232
Tile Horizontally 105
Tile Vertically 105

TIME OFFSET (MS) 50
Timer (モジュール) 406
TIME SYNC 50
Toggle 190
To Voice (モジュール) 414
TRANSPARENCY 198
Triangle 78
Triangle (モジュール) 312
Tri FM (モジュール) 312
Trigger 190
Triggered Random (マクロ) 258
Trim Selection 239
Tri/Par Symm (モジュール) 313
Tri Sync (モジュール) 313
Trp 222
TUNE 135, 220
TUNING 135
Tuning Info (モジュール) 418
TYPE 175

U

Undo 88, 108
UNISON 132
UNISON SPRD 135
Unison Spread (モジュール) 422
Unit Delay (モジュール) 389
Update 221
UPPER NOTE 143
USER CONTENT 98

V

VALUE 232
Value 4, 236
Value Auto Fit 236
Value (モジュール) 404

Vertical 200
VISIBLE 174, 176, 178, 236
Visible 236
VISIBLE IN VIEW A AND B 234
VOICE ALLOCATION 135
VOICE ASSIGN 136
Voice Info (モジュール) 417
VOICES 132, 135
Voice Shift 4
Voice Shift (モジュール) 415
V Scroll Bar 235
VST 25
VST プラグ・イン 37

W

Waveshaper (マクロ) 262
Wrap 231
Wrapper (マクロ) 262

X

X 231
X Auto Fit 235
X SIZE 231
X UNITS 232
XY 231
XY Scope (マクロ) 242
XY (モジュール) 277

Y

Y 231
Y Auto Fit 236
Y SIZE 231
Y UNITS 233

かな

あ

アキュムレーター 398

値変更 404

アフタータッチ 287, 292

アンサンブル 117

アンサンブル・ストラクチャー・ウィンドウ 119

アンサンブル制御パネル・ウィンドウ 84, 120

アンプ 308

アンプ系 244

アンプ (マクロ) 245

い

イコライザー (マクロ) 258

位置イベント 250

イテレーション 403

イベント処理系 263, 398

イベント信号 157

イベント・テーブル 407

インストゥルメント 129

インストゥルメント・ヘッダー 131

インストール手順 (MacOS X) 21

インストール手順 (Windows) 18

インテグレーター 382

インバーター (マクロ) 245

え

エンベロープ系 256, 352

エンベロープ生成器 406

エンベロープ (マクロ) 256

お

オーディオ→イベント変換 413

オーディオ→ゲート変換 414

オーディオ信号 157

オーディオ・テーブル 329

オシレーター 247

オシロスコープ 278

オシロスコープ (マクロ) 242

折れ線波発振器 326

音域情報 419

か

ガイガー・カウンター 329

開放 157

カウンター 398

拡散器 384

画像オブジェクト 279

仮想メモリー 211

加法 295

監視 (OSC) 51

き

基本ストラクチャー 153

基本マクロ 64, 145

逆数 297

逆正弦 303

逆正接 304

逆余弦 303

く

クオンタイザー (マクロ) 261

クオンタイズ 300

クライアント (OSC) 50

クリッパー 391

クリッパー (マクロ) 261

クロスフェード 306
クロスフェード(マクロ) 245
クロック 288, 293
クロック信号 327

け

計数器 398
ゲート 285, 291
ゲート信号 157
結線 66, 162
減算方式(シンセサイザー) 63
減法 296

こ

コア 3
コア・マクロ 64
コピー保護キー 7
コム・フィルター(マクロ) 258
混成モジュール 264
コンテキスト・メニュー 84
コントローラー 286, 292

さ

サチュレーター 390
雑音発生器 328
雑音(マクロ) 247
サチュレーター(マクロ) 262
三角波発振器 312
サンプラー系 248, 331
サンプラー(マクロ) 249
サンプリング 211
サンプル & ホールド 396, 404
サンプル & ホールド(マクロ) 258
サンプル・エディター 214
サンプル・マッパー 220

サンプル・マップ 214
サンプル・マップ・エディター 217
サンプル・ループ・プレーヤー 62

し

シーケンサー系 250, 349
シェイパー 393, 400
しおり機能 4
識別名(OSC) 49
指数 300
システム情報 418
試聴 115
主ツールバー 107
出力ポート 425
順序制御 402
衝撃波発振器 323
条件分岐 403
乗法 296
剰余 298
除法 297
信号源 159
信号受信 425
信号送信 425
信号配送系 305
信号分配器 307

す

スイッチ 271
数学演算系 295
スキャナー 305
スキャナー(マクロ) 247
スキン 5
スコープ 278
スタンドアローン 23, 27
ステップ・フィルター 404

ステレオ・パン 307
ストラクチャー・ウィンドウ 84
スナップショット 55, 201, 420
スムーサー 416
スライダー 266
スルー・リミッター 395

せ

制御信号源 159
制御パネル 171
正弦 302
正弦波発振器 316
斉奏 422
製品の認証 8
声部情報 417
積分器 382
絶対値 298
セバレーター 403
セクター 305

そ

ソング位置 289, 293

た

ターミナル系 425
帯域分割 (マクロ) 258
台形波発振器 321
対数 301
タイマー 406

ち

遅延 26, 383
チャンネル情報 419
チャンネル・メッセージ 289, 294
チューニング 418

調律 417
調律情報 418
チョッパ 393

つ

ツールバー 107, 109, 110

て

低周波発振器 352
低周波発振器 (マクロ) 257
定数信号源 161
定数値 295
ディストーション系 390
ディファレンシエーター 382
ディフューザー 384
ディレイ系 260, 383
ディレイ (マクロ) 260
テープデッキ 409
テキスト 276

と

同期クロック 288
同期処理 (OSC) 49
動的ポート管理 265
ドラム・マップ 216

な

内蔵プレイヤー 99

に

入出力端子系 425
入出力ポート 157
入力ポート 425
認証 8
認証キー 8

の

ノイズ 328

ノイズ (マクロ) 247

ノート・ピッチ 284, 291

鋸波発振器 309

ノブ 268

は

排他的論理和 402

階段波発振器 325

配置調整 (制御パネル) 75

波形モード 332

発振器系 247, 309

発振器 (マクロ) 247

発振器モード 331

パネル系 266

パネルセット 4

パネルセット・バー 121

パルス波発振器 318

パン 307

反復 403

パン (マクロ) 246

ひ

ピーク・デテクター 396

比較 299

ピクチャー 275

歪み系 261, 390

ピッチベンド 284, 291

ビットマップ 275

微分器 382

表示系 242

ふ

フィルター・エンベロープ 81

フィルター系 258, 367

フィルター (マクロ) 258

フェーダー 266

符号反転 296

ブラウザー 112

プラグ・イン 24, 32

プレイヤー 99

プログラム・チェンジ 287, 293

分周器 397, 399

へ

併合 404

平方根 302

冪 301

ペロシティー 285

ほ

ボイス 417

放物波発振器 314

ポート 130, 147, 157

ホールド 406

補助系 409

保存 (インストゥルメント) 76

ボタン 269

ま

マージ 404

マウス 83

マウス領域 281

マクロ 63, 241

マクロの切り替え表示枠 283

マスター・チューン 417

マスター・ボリューム (マクロ) 245

マルチ・テキスト 276

マルチ・ピクチャー 275

み

ミキサー 308

ミキサー系 244

ミキサー (マクロ) 245

ミラー 392

め

メーター 274, 275

メニュー 270

メニュー・コマンド 86

メンバー・リスト (OSC) 50

も

モジュール 153, 154

文字列 276

ゆ

ユニゾン 422

ユニゾン・スプレッド 422

よ

余弦 303

ら

ラダー (マクロ) 259

ラッパ (マクロ) 262

乱数生成器 399

乱数生成器 (マクロ) 258

ランプ 273

ランプ波発振器 327

り

リスト 270

両極パルス波発振器 322

量子化 300

リレー 306

リング変調器 (マクロ) 262

る

累算器 398

ルーター 405

ループ・エディター 225

ループ長 332

れ

レベル・メーター 275

レベル・ランプ 273

ろ

論理積 401

論理否定 402

論理和 401