

# The MOR MUSIC plug-in (Beta Version)

## Installation

Finale 2005 or 2006 must be installed on your computer before initiating the installation of MOR Music Plug-in.

## Introduction

The MOR Music plug-in is comprised of two applications:

- MOR Music **Main application**
- MOR Music **Preset Editor**

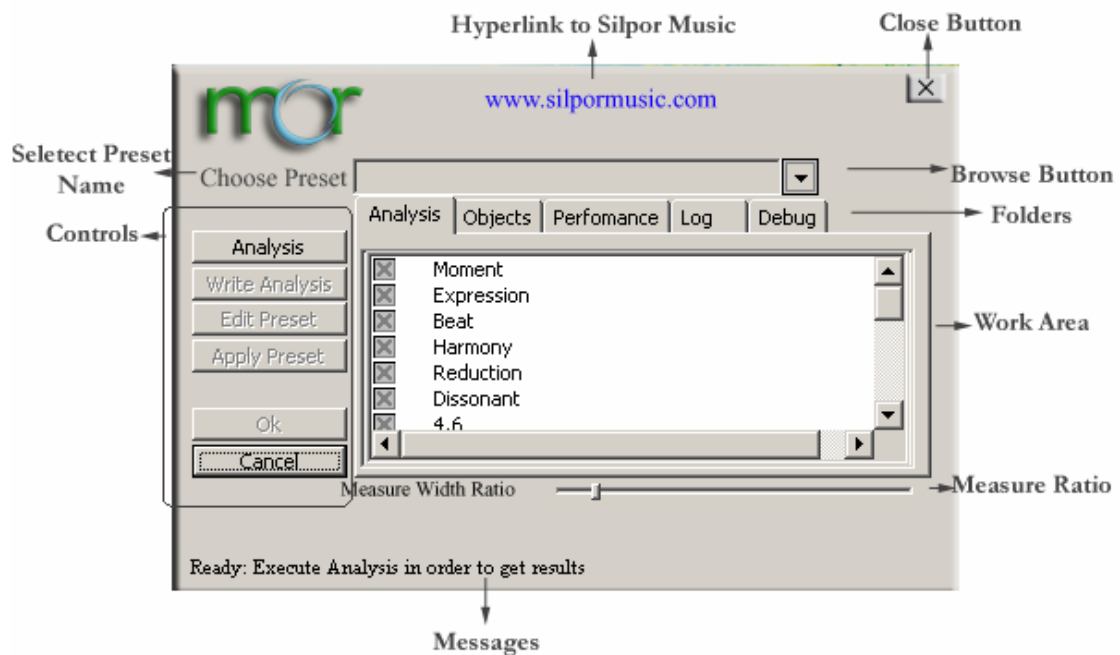
The **Main application** performs two functions:

- **Analysis**
- **Preset Application**

The **Analysis** is the process which scans the score and identifies the inherent music objects.

The **Preset application** attaches user defined parameters processors to the object identified by the analysis.

## The Main Application



## The main application controls

**Hyperlink to Silpor Music:** Click to get to Silpor Music's website for news and information about product updates, promotions, help and more.

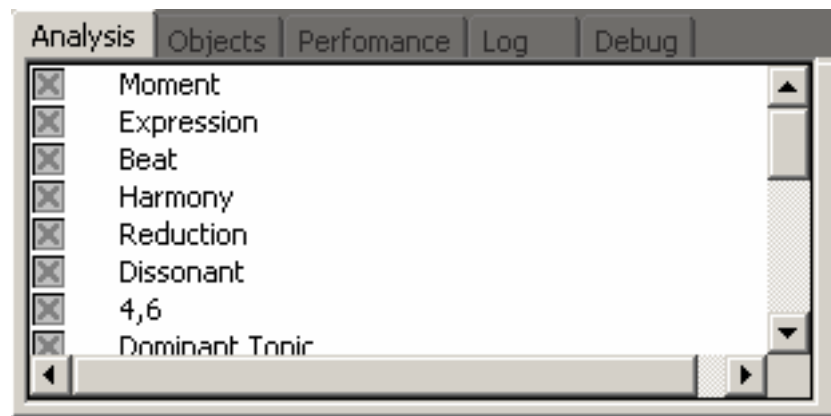
**Close Button:** Click to exit

**Choose preset:** Installed presets are accessed via the down-arrow to the right of the preset name display..

**Folders:** The various processes are organized in five folders.

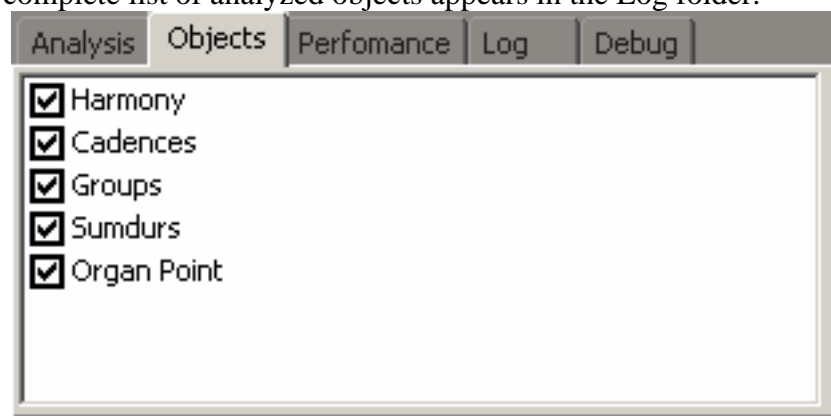
### Analysis

This is the process of searching, identifying and delimiting music objects. A progress bar tracks the progress of the analysis of each object.



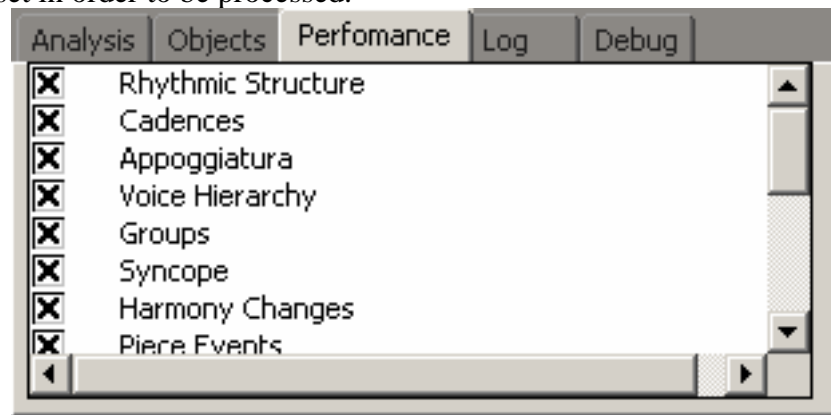
### Objects

When the analysis is completed, this folder displays graphically some of the analyzed objects. Please note that all analyzed objects are employed during the Performance process. This list is selectable and you can choose to display as many objects as you wish by selecting them and pressing the 'Write Analysis' button. A complete list of analyzed objects appears in the Log folder.



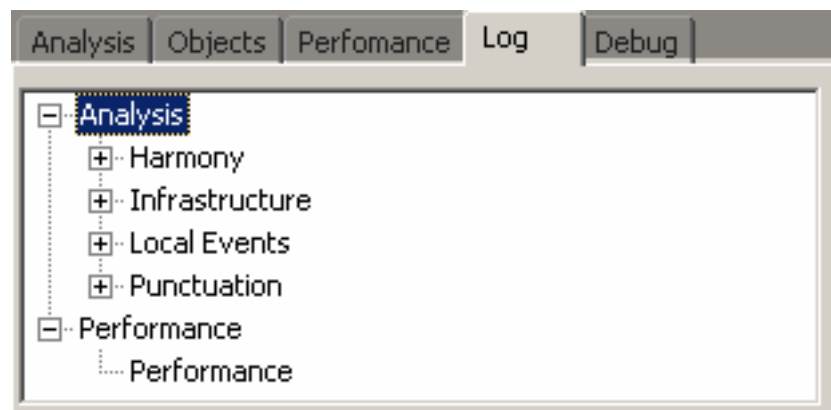
### Performance

This folder lists all the objects available for parameters values determination in a preset. The list is selectable and you can choose as many objects as you wish for processing by the preset. Please note that the object must be defined in the preset in order to be processed.



### Log

This folder contains all the log files of the saved path of the analysis and the performance sorted according with the process has created. Object can be viewed by opening the correspondent key and choosing the referred object. Please note that all the Performance processors are saved under the key "Performance"



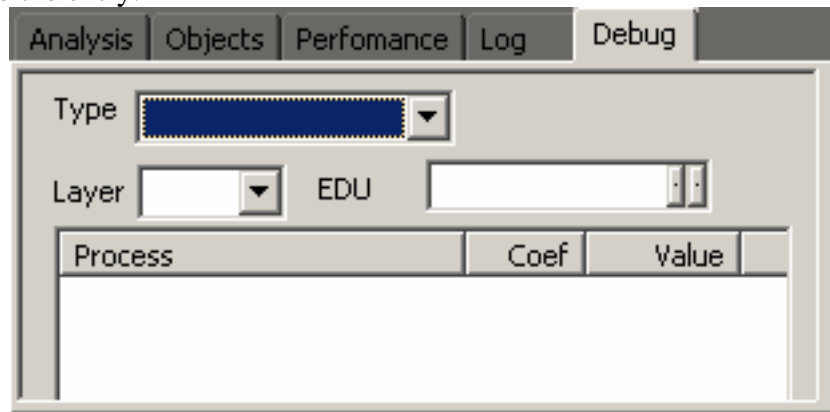
### Debug

This folder provides detailed description of processes affecting any note or moment in the score. You can select

- the performance parameters you want to debug, for instance, Velocity, Tempo, Duration.
- the layer you want to see (if applicable), and
- the position you want to see in EDUs. (EDU stands for Elapsed Duration Units, where a quarter-note is equal to 1024 units).

By clicking on the arrows to the right of the EDU text box you can scroll the EDUs forwards or backwards. Each EDU that has at least one process attached

to it would appear in the table. By double clicking on any entry in the table, the correspondent Process Editor will open displaying the current preset attached to the entry.



**Measure Ratio:** Enables adjusting the Finale measure width to prevent overlapping texts, when the 'Write Analysis' is deployed in the 'Objects' folder, especially when writing the Harmony analysis. By adjusting the measure ratio, the default width of the measures may be expanded or compressed.

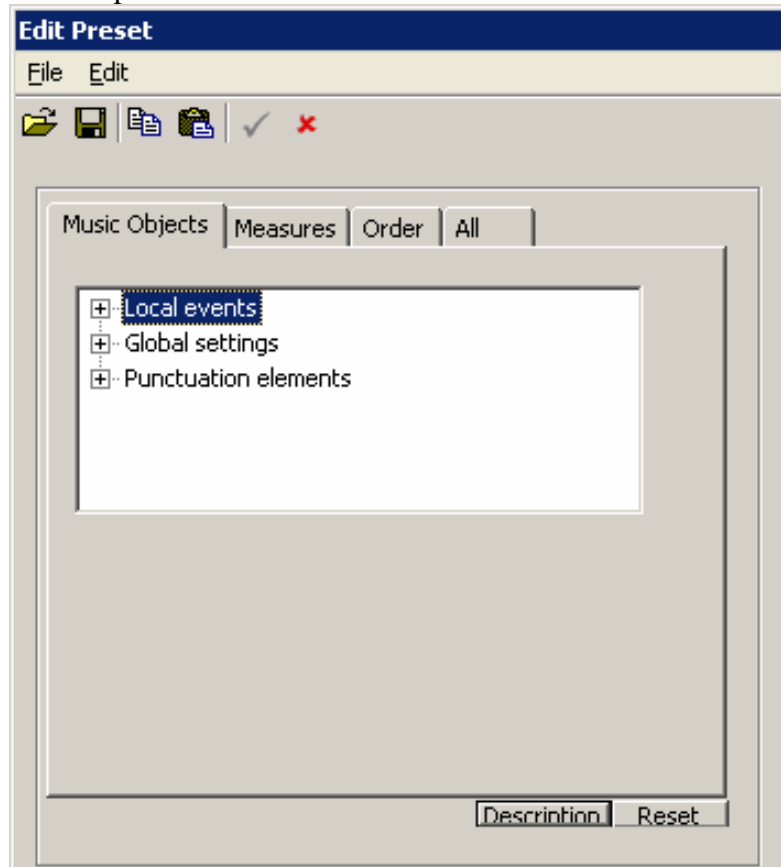
**Messages:** the system prints messages announcing the current status of processes performed.

**Controls:**

- Ok: Exits the application saving changes made by the performance.
- Cancel: Exits the application without saving the changes made by the performance
- Apply preset: Applies the selected preset process on the current score.
- Edit Preset: Opens the 'performance' folder for modifying the values of the current preset.
- Process: Currently not in use.
- Write Analysis: Graphical displays selected objects.
- Analysis: Performs the analysis process.

## The Preset Editor

The 'Edit Preset' button in the controls of the main application opens the editing dialog for a selected preset.



## The Preset Structure

Editable music objects are organized according to the following classifications:

- **Music Objects:** Objects that contain more than one note and have a definable start and end.
- **Measures:** Objects delimited by a specified single measure.
- **Order:** The order in which processors are deployed in the preset
- **All:** a list of all the processors engaged in the preset

## Music Objects

The Music Objects are grouped into a hierarchic tree:

- **Local Events:** Objects that affect only the area where they are localized. They may or may not appear in the piece.
- **Global Settings:** Objects that affect the overall structure of the performance. They must be present.
- **Punctuation elements:** Objects that enmark phrasing areas of the piece.

Subgroups that they are common in all these groups are:

- **Score Marking:** Objects that are represented explicit in the piece.
- **Analyzed:** Objects that are deduced by the analysis algorithm.
- **Tempo:** Objects that generally affect the tempo of the piece.
- **Velocity:** Objects that generally affect the velocity of notes involved.
- **Change:** Objects that change the actual value for another.
- **Dynamic:** Objects that generally indicate a gradual change from one value to another.
- **Setting:** Objects that generally indicate a new value for a specific point of the piece.

## Measures

In a process called **Measure Rhythmic Structure**, each measure is assigned a graph for Tempo and a graph for Velocity defining values for each moment in the measure in accordance with the meter of the measure and the subdivision of the notes present in the measure.

Tempo modifications of moments within a measure can also be made in accordance with the frequency of harmony changes, in process called **Chord Flow**.

## Processors

The following is a list of the available types of processors:

**Set up:** A number which changes the current value of an object's parameter.

**Set up Graph:** A graph which changes the current value of an object's parameter

**Factorized Graph:** A graph which changes the current value of an object's by a factor value rather than a static value.

**Groove Graph:** A column graph providing an exact or factorize value for an exact position.

**Cartesian Axes:** A graph, which considers two parameters for determining the value, for example Tempo and Velocity.

**Ad-hoc Processors:** Processors designed for specified objects. For example, the piano's sustaining pedal.

The following table lists currently available processors:

<b>Processor</b>	<b>Description</b>
Velocity	<b>Set up</b> a new velocity
Velocity Graph	<b>Cartesian graph:</b> x = Time y = Velocity This graph can be designed as a factorized or a set-up graph.
Compress Velocities	Two values are required. This processor will compress the current velocities within a user determined range.
Velocity by Pitch	<b>Cartesian Graph:</b> x = Pitch y = Velocity This graph can be designed as a factorized or a set-up graph.
Velocity by Duration	<b>Cartesian Graph:</b> x = Duration in ms. y = Velocity This graph can be designed as a factorized or a set-up graph.
Velocity Factor	Apply a factor value to the current range.
Tempo	<b>Set up</b> a new tempo for the object.
Compress Tempo	Two values are required. This processor will compress the current tempi within a user determined range.
Tempo Factor	Apply a factor value to the current range.
Duration	<b>Set up</b> a new duration for the object
Articulation	<b>Cartesian Graph:</b> x = Duration in ms. y = new Duration This graph can be designed as factorize or set-up.
Duration Factor	Apply a factor value to the current range.
Add Duration	Add a specified duration to the current duration of the object.
Pedal Controller	Special controller

### **Range determination**

Once an analyzed object and a processor are attached, the processor, by default, operates within the limits of the object. The following table lists optional range determination:

<b>Range</b>	<b>Description</b>
By number of beats	A value indicating the number of beats before or after an event, applied only to expressions.

By number of notes	A value indicating the number of notes before or after an event, applied only to expressions.
By Object rules	(Default) according to the defined object boundaries.
By range (EDU)	The range in EDUs and the layer(s) upon which the process operates.
Single Chord	The process affects all the notes present at the event's starting moment.
Single Note	The process affects only the specific note comprising the event starts.
Subdivision	A value indicating the number of moments before or after an event.
Time Limit (ms.)	A value indicating the time in ms the before or after an event.
Until end of Piece	The process will be applied to all the notes following the event until end of the piece.
Until next event	The process will be applied to all the notes up to the appearance of the next event.

### Filters

Filters offer additional user control over the operation of the processors.

Following are currently available filters:

Filter	Description
Channel Filter	The process affects only the listed channel(s).
Duration Filter	The process affects only the notes, which their duration in milliseconds matche the range defined in the filter.
Moment subdivision	The process affects only the moments, which their subdivision in EDUs match the range defined in the filter.
Notes Subdivision	The process affects only the notes, which their subdivision in EDUs match the range defined in the filter.

### The editor controls

**Merge:** The 'merge' option is intended to moderate the cumulative effect of multiple processors operating on a single object.

**Copy:** This button copies the definitions of the current process to the clipboard for pasting into another similar process. Copying a process into a different kind of process is not allowed.

**Paste:** The counterpart of the Copy option. Pastes process definition from the clipboard into the current process.

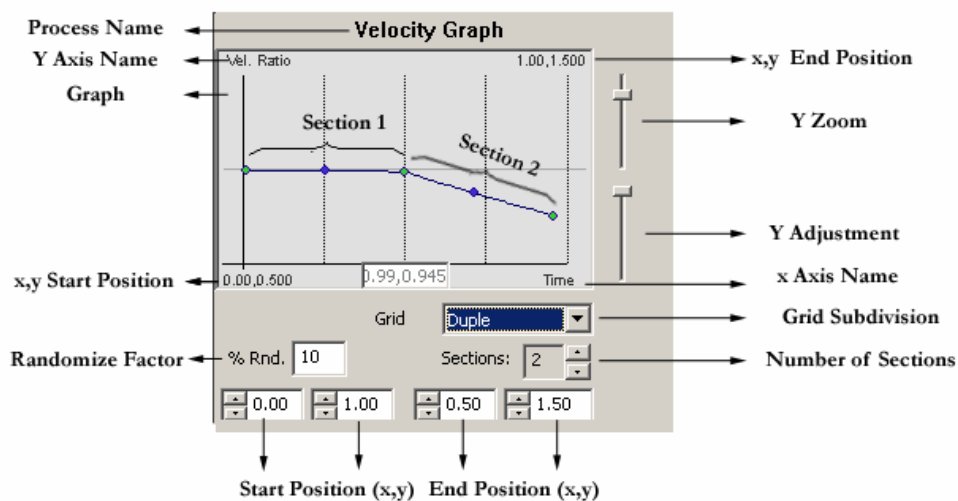
**Load:** Import a previously saved process to the system.

**Save:** Export the current process out of the system.

**Ok:** Accept the current modifications.

**Cancel:** Abort the current modifications.

## Working with graphs



### Grid

Optionally you can choose the correct grid subdivision fitting the piece meter.

For example, 3/4 and 6/8 fit a triple grid subdivision, or 2/4 and 4/4 fit a double grid subdivision.

### X,Y Axis

The X-axis is normalized to 0-1 range, and in certain cases the Y-axis is normalized to 0-1 range as well.

When an Axis is normalized to 0-1 range, the entire range of X is represented in values from 0 to 1, where 0 is the minimal and 1 is the maximal value.

For example, if an object is found between the EDU 1024 and EDU 2048, 0 would be EDU 1024, and 1 – EDU 2048. Thus, the delta value of 1024 (2048 – 1024) would be represented as a curve, in which each point equals 1/1024.

### Graph Sections

The graph is comprised of a sequence of curves concatenated at the start/end points.

Each curve has three control points: Start, End point (green) and the curve control point (blue). To move the start or end point drag and drop the green points. To change the convexity of a curve, drag and drop the blue point.

Sections can be added for increased precision.

## Graph Adjustments

The graph has three adjustment parameters: x,y Start Position, x,y End Position and Y Adjustment.

- **x,y Start Position:** The minimal value represented in the graph
- **x,y End Position:** The maximal value represented in the graph.
- **Y Adjustment:** Y multiplier where 1 is the default value.

## The Randomize factor

Introduces error deviations into the curve in accordance with the value entered .

## Factorize Option

The graph has two types of results: Set values and factorized values. (Use the scroll bar to view these options).

If the values in the Y Axis of the graph are Set Values, these values are applied to the object.

If the result of the graph is a coefficient to be applied in the object, the graph is factorized.

Note that Set values will override the old value of the object with the current graph value.

## Working with groove graphs

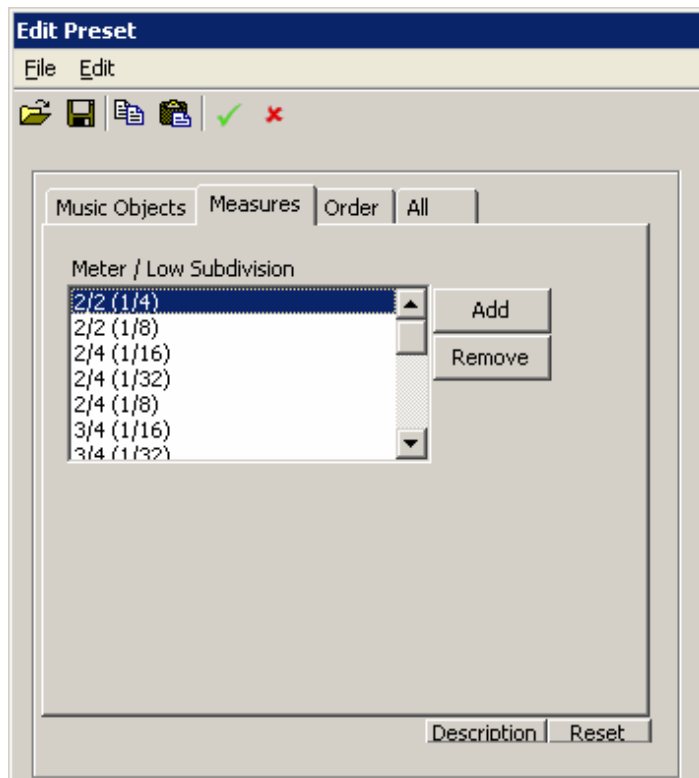
The following score will serve as an example for working with Tempo grooves and Velocity grooves in the 'measures' classification.



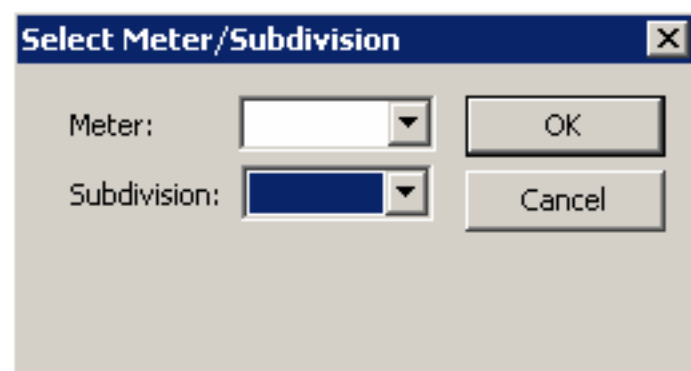
The score is comprised of three measures in 4/4 meter. Each measure has a different distribution of subdivisions. The grey vertical lines form the boundaries of 'moments', which define the appearance of notes in the timeline. Each moment starts with the appearance of a note and ends with the appearance of another note.

In the first measure there are 4 moments each being a quarter-note. In the second measure there are 6 moments: the first being a quarter-note, the second and third – eighth-notes, the fourth – a quarter-note, the fifth and sixth – eighth-notes. In the third measure there are 7 moments: one quarter-note, 4 sixteenth-notes and 2 more quarter-notes.

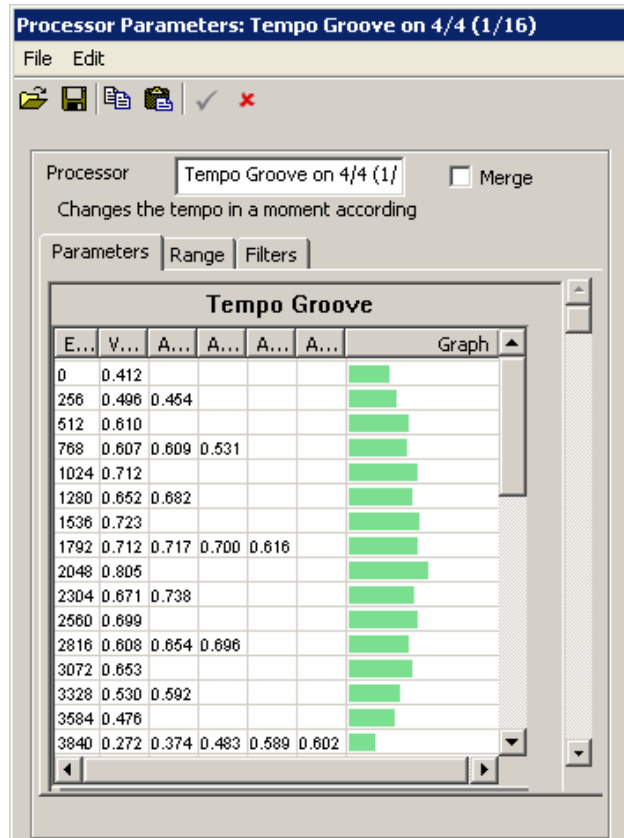
In order to apply a processor to each measure, a processor corresponding to the minimal subdivision of each measure must be defined.



In the above screen shot the three processors required for the example are already defined. To add processors press the 'Add' button in order to open the Meter/Subdivision definition dialog and choose a meter from the Meter list and a subdivision from the subdivision list.



To define a Measure Rhythmic Structure; choose from 'measures' the type of measure you want to define and select a **Groove** processors (Tempo groove, Velocity groove)



The above screen shot is a graph assigning tempo to moments is a measure in 4/4 time with a minimal subdivision of sixteenth-notes.

The first column is telling us the EDU value for row.

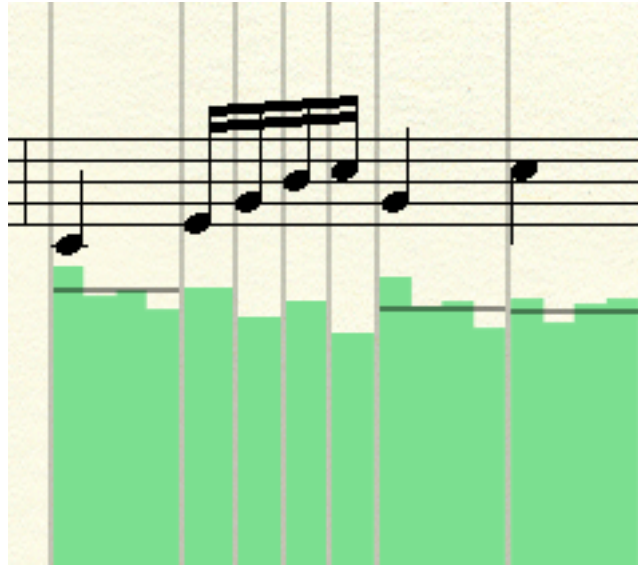
The second column is the value for the row.

The following columns are describing the average values for this meter and subdivision, from the smallest unit to the larger.

For example: in 4/4 meter with a subdivision of 1/16, the smallest unit is two sixteenths notes together. Therefore, we have 8 averages which is 16/2. The next average will be each 4 notes, then each 8 notes and the largest must be each 16 notes, which is the whole measure.

The last column is a graphic representation for each value.

The result of applying the groove graph in a measure containing different type of notes is represented in the following graph:



When one moment is longer than the subdivision defined in the groove, the system will take all the subdivision values that fit this moment, and will calculate an average value of them, which in this graph is represented by the horizontal lines.

Notice: Groove processors will not affect measures whose properties have not been defined.

## Working with the piano sustaining pedal

Harmony Pedal

Set Pedal to  
127

EDU Harmony Start  
From 0 To 0

EDU Harmony End  
From 0 To 0

% of Dissonant Start  
From 5 % To 10 %

% of Dissonant End  
From 5 % To 10 %

Harmony Pedal On  
127

Harmony Pedal Off  
0

Dissonant Resolution Pedal Off  
127

Dissonant Resolution Pedal On  
16

On/Off travel  
64

Minimal Hold Time  
64

The above screen shot is the pedal processor dialog.

To activate the pedal processor at least one pedal expression *Ped.* must be present in the score.

There are two types of pedal processors available:

- Pedal according to harmony analysis
- Pedal according to score expressions

The pedal according to harmony analysis applies the pedal in accordance with the harmony changes the dissonant analysis of the score. The pedal according to score expressions applies the pedal according to the 'Pedal Start' and 'Pedal End' markings in the score.

### Pedal Parameters

- **Harmony Pedal:** Tells to the system (when checked) to apply the pedal processor in accordance with harmony analysis
- **Set Pedal To:** The MIDI value for 'Pedal On'
- **EDU Harmony Start:** The offset of the 'Pedal On' in the Harmony Pedal.

- **EDU Harmony End:** The offset of the ‘Pedal Off’ in the Harmony Pedal.
- **Dissonant Start:** The offset of the ‘Pedal Off’ when a Dissonant appears in ‘Harmony Pedal’
- **Dissonant End:** The offset of the ‘Pedal On’ when a Dissonant disappears in Harmony Pedal.
- **Harmony Pedal On:** The MIDI value for ‘Pedal On’ when the harmony changes in ‘Harmony Pedal’.
- **Harmony Pedal Off:** The MIDI value for ‘Pedal Off’ when the harmony changes in ‘Harmony Pedal’.
- **Dissonant Resolution Pedal On:** The MIDI value for ‘Pedal On’ when the resolution of a dissonant is found in ‘Harmony Pedal’.
- **Dissonant Resolution Pedal Off:** The MIDI value for ‘Pedal Off’ when the resolution of a dissonant is found in ‘Harmony Pedal’.
- **On/Off Travel:** The time in EDUs that the Pedal takes to change position from ‘On’ to ‘Off’ or from ‘Off’ to ‘On’.
- **Minimal Hold Time:** This option is designed to prevent negative pedal activation durations or extremely short pedal activation durations.

If the “Until Next Event” range is applied to the pedal processor a ‘Pedal Off ‘ mark or a new ‘Pedal On’ mark must be present.

## The processors order

Processors are applied in accordance with a user defined order index.

This index tells the system in which order to deploy the processors.

As a general processors defined by set-up values should be deployed before processors defined by factorized values. In addition, processors without the ‘Merge’ option must come before processors with the ‘Merge’ option.

To change the order of the processors select the process and use the Up/Down arrow to reposition the process.

## Debugging

There are three ways to debug the results:

- **Graphic Representation:** By utilizing the **Write Analysis** button you can view a graphic display of objects such as Cadences, Modulations, Organ Points, Harmony and Groups.
- **Log Files:** By opening the log file correspondent to a processor or analysis you can verify whether the processor worked as expected.
- **Debug Dialog:** By choosing the type of event (Tempo, Velocity, Duration, Pedal) and the position in the score in EDUs, you can view the processors deployed in this position. This dialog works only with performance results, therefore a preset must be applied prior to accessing this dialog.

## Finale Issues

### Smartshapes

The MOR System will see a smartshape, which has been attached to a staff, as belonging only to that staff. For example, hairpins will affect only the staff they are attached to.