


PatchWork

Kant

Version 1.5

A d d e n d u m

First English Edition, March 1996

IRCAM  Centre Georges Pompidou

This manual may not be copied, in whole or in part, without written consent of Ircam.

This manual was written by Carlos Agon et Gérard Assayag, translated into English by J. Fineberg, and was produced under the editorial responsibility of Marc Battier, Marketing Office, Ircam.

Patchwork was conceived and programmed by Mikael Laurson, Camilo Rueda, and Jacques Duthen.

The Kant library was conceived and programmed by C. Agon, G. Assayag, and C. Rueda, with musical expertise provided by J. Fineberg.

First English edition of the documentation, March 1996.

This documentation corresponds to version 1.5 of the library, and to version 2.1 or higher of PatchWork.

Apple Macintosh is a trademark of Apple Computer, Inc.

PatchWork is a trademark of Ircam.

Ircam
1 place Igor-Stravinsky
F-75004 Paris
Tel. (33) (1) 44 78 12 33
Fax (33) (1) 42 77 29 47
E-mail : ircam-doc@ircam.fr

IRCAM Users' group

The use of this software and its documentation is restricted to members of the Ir-cam software users' group. For any supplementary information, contact:

Département de la Valorisation
Ircam
Place Stravinsky, F-75004 Paris

Tel. (1) 44 78 49 62
Fax (1) 42 77 29 47
E-mail: bousac@ircam.fr

Send comments or suggestions to the editor:
E-mail: bam@ircam.fr
Mail: Marc Battier,
Ircam, Département de la Valorisation
Place Stravinsky, F-75004 Paris



To see the table of contents of this manual, click on the **Bookmark Button** located in the **Viewing** section of the **Adobe Acrobat Reader** toolbar.

Contents

Résumé 6

Important Notice 7

Overview of Additions and Modifications 8

- New Inputs 8

- obj Input 9

- bpf Window 9

- Patterns 10

- Standard Segmentation 11

- Alignment to the Right or to the Left 11

- Options _4 and _8 11

- Max Relative and Global 11

- Local Control of the Parameters precision et max/ 12

- User Defined Segmentation Methods 13

Index 16

Résumé

Ce manuel présente les nouvelles fonctionnalités offertes par la version 1.5 de la librairie Kant de PatchWork. Il doit être utilisé en complément du manuel de référence « PatchWork Kant - Rhythm Quantification Editor - User's manual ».

Les changements apportés à Kant peuvent causer la perte de l'information de quantification lorsqu'on utilise un patch réalisé avec l'ancienne version. Le patch, toutefois, demeure utilisable.

1 Important Notice

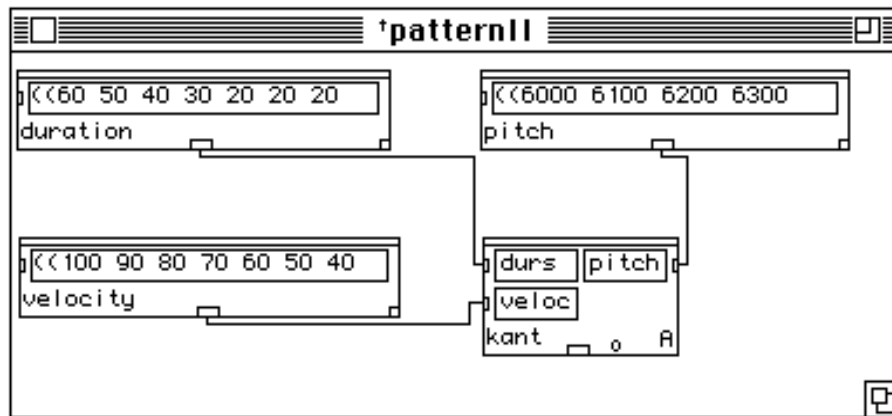
This manual presents the new feature and the modifications to existing features of the library Kant. It is intended to serve as a complement to the manual "PatchWork Kant - Rhythm Quantification Editor - User's manual".

The changes made to Kant can cause the loss of the quantification editing information contained within a patch that was realized with the old version; the patch itself, however, will remain usable.

2 Overview of Additions and Modifications

New Inputs

Two new inputs have been added to Kant. These inputs permit the use of pitch and velocity information in the sequence being treated.

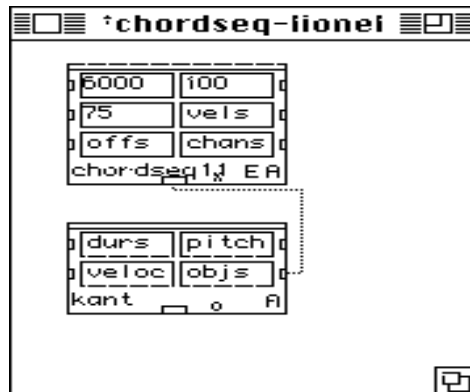


In principal the three lists connected to Kant's inputs should be of compatible length. If this is not the case the module will behave in the following manner :

- The basic list is that of durations.
- If one of the additional lists (pitch and velocity) is absent that information will be added using the default values (6000 for pitch and 100 for the velocities).
- If one of the additional lists contains more elements than the duration list, Kant will use the values of the list, in order, for as many elements as needed then ignore the rest.
- If one of the additional lists contains fewer elements than the duration list, the missing values will be supplied by the default values.
- In the case of rests (represented by convention as negative durations), there is no need for a corresponding pitch or velocity value and thus the additional lists must not contain them.

obj Input

Option-clicking on the box will cause a new input called *obj* to appear. This input may receive musical objects coming from editors such as **chordseq** or **multiseq**.

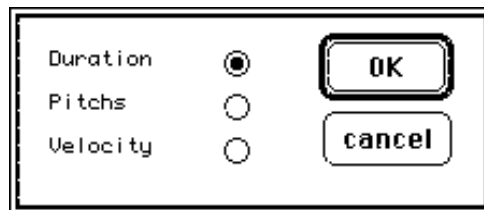


We impose the limit on these objects that, inside of the editors, each voice is monophonic (no chords or overlapping notes). If this is not the case, Kant will consider only the attacks of the various events. If there are chords, Kant will keep only the lowest note and eliminate the rest. In order to treat polyphony, you must construct multiple monophonic sequences in a **multiseq**.

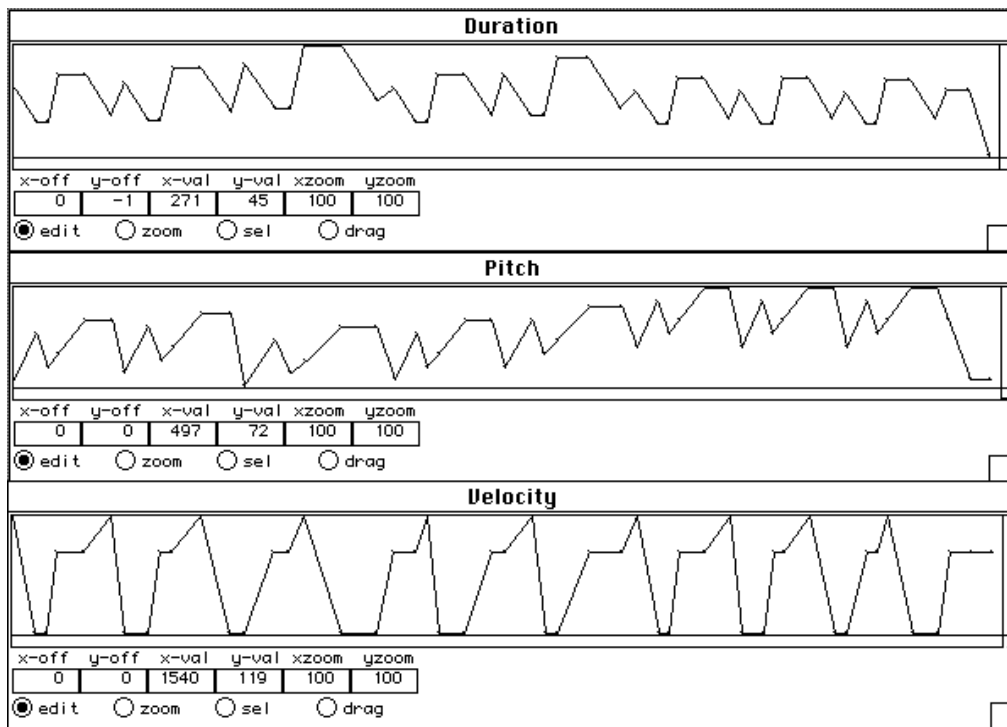
If the *obj* input is used, the other inputs will be ignored.

bpf Window

Once in the Kant editor, we have enlarged the graphic representation to allow the display of pitch and velocity. By selecting the option **bpf** from the **archi** menu you will see the following dialogue box:



This gives access to simultaneous graphical representation of one two or all of these parameters. The editing commands for the new windows are identical to those of version 1.0.



Patterns

With the introduction of pitches and velocities, segmentation by motives becomes an important method of segmentation for use in Kant.

- The usable symbols {> < =} have been enlarged to include the new symbols <p >p =p and <v >v =v for notes (pitch) and velocities respectively
- The new symbol ? has been introduced as a wild-card to refer to event of indiscriminate character.
- It is also possible to use the syntax of PatchWork's expand-list to describe your motives in an abbreviated fashion (see the manual PatchWork Introduction).
- The operator and has been introduced, since it is now possible to request conditions in different domains.

Example :

```
(( ( < + ) ( ? + ) ( > + ) )
 ( < v + )
 and)
```

This example recognizes a pattern in which the rhythm accelerates, the performs indiscriminately, then slows, during all of this the velocities must be decreasing.

Standard Segmentation

The idea behind this new method of segmentation is very simple: you can locate events within the sequence that satisfy the constraints imposed by the following dialogue box:

duration	<input checked="" type="radio"/> equal	<input type="text"/>	<input type="radio"/> between	<input type="text"/>	and	<input type="text"/>
pitch	<input checked="" type="radio"/> equal	<input type="text"/>	<input type="radio"/> between	<input type="text"/>	and	<input type="text"/>
velocity	<input checked="" type="radio"/> equal	<input type="text"/>	<input type="radio"/> between	<input type="text"/>	and	<input type="text"/>
						<input type="button" value="OK"/> <input type="button" value="cancel"/>

Two types of conditions may be imposed on the different types of events. By using the `equal` button it is possible to find event whose value is equal to a given value. If the `between` button is used, the selected events will fall between the two limits given in the inputs to the right of the button.

Alignment to the Right or to the Left

The parameter edge which positioned the segmentation to the right or to the left has been eliminated from all the methods. In order to move the different segmentation to the right or left, select the method by double clicking on one of its marks, then use the characters `→` or `←` depending on the case.

Options `_4` and `_8`

The options `K1` and `K2` have been replaced by the options `_4` and `_8` in the dialogue box `recherche de tempo` (menu `measure>go`). With the option `_4` the unit of pulsation is always the quarter note. With `_8`, it may switch back and forth between quarter notes and eighth notes depending on what Kant considers optimal for each measure. In version 1.0 Kant decomposed measures based on 8 as the addition of a measure in $n/4$ plus a measure in $1/8$; for example, if we had a measure in $5/8$ it would have been written as a measure in $2/4$ followed by a measure in $1/8$. In version 1.5, measures based on 8 are correctly notated (i.e. $5/8$)

Max Relative and Global

By using the option `_8` we will often have a metric which moves back and forth between measures with a pulsation in 4 (quarter notes) and measures in 8 (eighth notes). This can pose a problem in setting the value for the parameter `Max` (the maximum number of allowed subdivisions of the beat, menu `quantify>global`). Suppose we set the value of `Max` at 8 : this implies that for a measure with a quarter note pulse the maximum division is the 32nd note, while in a measure with an eighth note pulse the maximum division would be a 64th note. How-

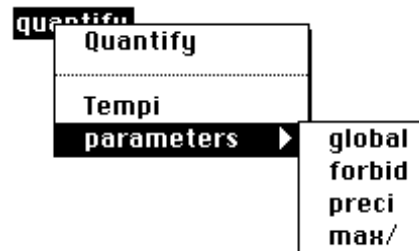
ever, that same value of 8 for Max might also be used to mean that the maximum division everywhere is the 32nd note (in reference to a quarter note). Since both points of view are valid and useful we have introduced an option for the parameter Max :

Max-T ☒ Max-R ☐
 Preci
 Forbid

If you chose the option Max-T (*Total*) you will allow a maximal subdivision of 8 relative to a quarter note. Alternatively, if you chose the option Max-R (*Relative*) the maximum value of subdivisions will be determined measure by measure, relative to each measure unit of pulsation.

Local Control of the Parameters precision et max/

As with the parameter **forbid**, it is now possible to control the values of **precision** and **max/** for a single measure or even a single beat. The Parameters menu has been enlarged for these operations :



When editing the value of one of these parameters in local mode (by clicking directly on the measure to be edited) the syntax is the same as for the parameter **forbid**. The key-word **def** has been added to refer to the value that was set for the parameter at the global level. For example, if you have set the global value for the parameter **forbid** to (6 7) and you make a local adjustment (in the first measure, for example) of this same parameter to the value (def (! 5) def), the first beat of the measure will follow the global setting (sextuplets and septuplets forbidden), the second beat will be forced into a quintuplet division, the third beat and the rest of the measure will go back to the global setting. It is also possible to integrate **def** in an arithmetic expression : if **max/** global is equal to 8, the expression **def / 2** allows the application of a **max/** local equal to 4.

User Defined Segmentation Methods

Kant allows the user to define personalized segmentation methods, to place them within the segmentation menu and to specify the segmentation parameters which will be accessible to the user as a dialogue box when the method is selected from the menu.

In order to add a custom method, you must add the text of its code to the end of the file `users-kant.lisp` which is located in the folder Kant. The definition must follow the following outline:

```
(defmethod my-personal-method
  ((self C-MN-view-mod-kant) ev-list arg1 arg2 ... argn)
  ... body of the method ...
)
```

The terms which are not in italics must figure exactly as shown.

The arguments *arg1 ... argn* are optional parameters which can be used in the body of the method. *my-personal-method* is the arbitrary name that you will chose for your method. `ev-list` contains the list of events to be quantified. You can use macros to access the attributes of each of these events :

<code>(dur event)</code>	retrieves the duration in 1/100ths of a second
<code>(pitch event)</code>	retrieves the pitch in midicents
<code>(velocity event)</code>	retrieves the velocity
<code>(onset event)</code>	retrieves the attack time in 1/100ths of a second
<code>(silence event)</code>	returns the value true if the event is a rest
<code>(voice event)</code>	returns the voice number of the event

The method must return a list of whole numbers which represent the index numbers inside the sequence `ev-list`. This sequence will then be segmented at the positions defined by those indexes.

Following the segmentation method, an expression in the following form is necessary :

```
(my-new-method-seg '(menu-name method-name symbol parameter-list))
```

<i>menu-name</i>	a string of characters which give the name that will appear in the segmentation menu.
<i>method-name</i>	name of the method defined by <code>defmethod</code>
<i>symbol</i>	a string with a single character, representing the symbol which will be shown in the Kant window at each segmentation point.
<i>parameter-list</i>	a list of pairs (parameter-name default-value). The names must correspond to the parameters <i>arg1 ... argn</i> of <code>defmethod</code> .

Example : segmentation by a threshold of durations, pitch and velocity

```
(defmethod method-threshold
  ((self C-MN-view-mod-kant) ev-list d-seuil v-seuil p-seuil)
  (let ((index 0) (segment-list nil))
    (dolist (event ev-list)
```

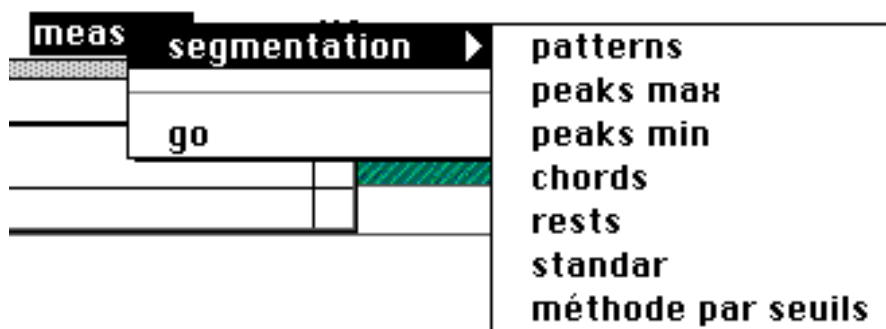
```

      (when (or(> (dur event) d-seuil)
              (> (pitch event) p-seuil)
              (> (velocity event) v-seuil))
            (push index segment-list))

      (incf index))
    (reverse segment-list)))
(my-new-method-seg
 '("threshold method" method-threshold  "®"
   ((d-seuil 30) (v-seuil 64) (p-seuil 7200 ))))

```

This example allows the user to obtain the following menu :



the following dialogue box :

and to display the segments with the symbol «®»:

kant

Ⓜ Ⓜ Ⓜ Ⓜ

509 stcnt

archi **measure** quanti

zoom

Index

> 10

A

Agon, C. 2
Alignment 11
archi 9
Assayag G. 2

B

between 11
bpf 9

C

chordseq 9

D

def 12
Duration 8, 13
Duthen J. 2

F

Fineberg J. 2
forbid 12

L

Laurson M. 2

M

Max 11
max/ 12
measure>go 11
multiseq 9

O

obj 9
Options
_4 11
_8 11
K1 11
K2 11

P

Patterns 10
Pitch 8, 10, 13
precision 12

R

Rueda C. 2

S

Segmentation 11, 13
Symbols

10, 10
=p 10
=v 10
>p 10
>v 10
? 10
® 14

V

Velocity 8, 10, 13

