

PATCHWORK newsletter

n° 4 - October 1996

This is the Newsletter for PatchWork. It is composed of frequently Asked Questions and Update lists for PatchWork. It serves two purposes:

- to reduce the number of "often asked questions" about PatchWork, by posting the questions and answers in this document so that they are shared by all.
- to inform the community of PatchWork users about updates and improvements.

Please send your corrections, questions, and comments to the editor, Marc Battier, at ircam-doc@ircam.fr.

Please report all bugs to: Mikhail Malt:

E-mail : ircam-doc@ircam.fr

Mail Ircam, 31, rue Saint-Merri, F-75004 Paris (France)

Fax 01 44 78 15 60

Copyright © 1996, Ircam

TABLE OF CONTENTS

Environment	2
New Developments	2
Running PatchWork.....	2
Installation	2
Launching PatchWork	2
Hardware and software requirement	2
Working with MIDI	2
AppleScript	3
Tutorials and documentations	3
Libraries	3
On-line doc and tutorial	4
What's new in PatchWork 2.6.3 ppc?	4
Module that have changed: Play-object ...	4
New PatchWork Modules :	5
midi-record	5
midi-load	5
midi-save	6
Libraries updates.....	6
PWConstraints	6
PW-MIDI Manager	6
Libraries current versions	7

Environment

New Developments

This newsletter corresponds to the release of PatchWork 2.6.3 for PowerMacintosh.

Running PatchWork

PatchWork is a visual programming environment in MCL 4.0 that lets you do whatever you might do in Lisp, but in a purely graphical way. It has many modules and libraries specialized for making, playing and notating music, but it can be used for any other purpose, like teaching and learning functional programming in Lisp.

Installation

Just copy the `PatchWork 2.6.3 ppc` folder to your hard disk.

If you already own MCL 4.0, it is wise to have the MCL folder at the same level than the `PatchWork 2.6.3 ppc` folder, so PatchWork can locate it. If PatchWork cannot locate MCL 4.0, the first time you ask for the documentation of a PatchWork module, it will issue a dialog "The help file could not be found. Would you like to locate the file?". Just answer "no", then everything will work fine.

Remember: do not change the location of any file that resides in the `PatchWork 2.6.3 ppc:Images` folder.

Launching PatchWork

To launch PatchWork, double-click on the file `PW2.6.3.image` in the folder `PatchWork 2.6.3 ppc:images`. Because of the protection system, make sure you understand the first installation procedure before launching PatchWork 2.6.3 for the first time.

Hardware and software requirement

- A Macintosh Power PC
- System 7.5.2 or higher. When using System 7.5.2, make sure that the Shared Library Manager extension file is in the system's extensions folder.
- A minimum of 8 MB of RAM for PatchWork itself, plus 4 MB that will be added to the system due to MCL 4.0 shared libraries. A RAM setting of 12000 MB reserved for PatchWork should be enough.

There is nothing to install in the extensions folder, as the MCL shared libraries have been included inside the image.

Working with MIDI

In order to take advantage of PatchWork MIDI capabilities, you must connect a MIDI interface to the modem port of your Mac.

PatchWork 2.6.3ppc includes Midi Share, a software developed by GRAME. MidiShare is a realtime multitasks music operating system specially devised for the development of musical applications. Its innovative features and careful design, result of 6 years of research and development at GRAME in Lyon, provide developers with a powerful and efficient toolbox for developing Midi applications. You can find lots of useful information on MidiShare and about GRAME's activities on their Web page, available in French and in English. Point your browser to: <http://www.grame.fr/>. You will also find an article published by Grame about Midi Share, in the Midi Share folder. It is called *MidiShare - un système d'exploitation musical pour la communication et la collaboration*, and was written by Dominique Foer, Stéphane Letz and Yann Orlarey. Read it with Adobe Acrobat Reader. There are also 2 "readme" files.

Midishare is simply installed by copying the MidiShare Control Panel to your Control Panel folder in the System folder. MidiShare is compatible with OMS. If you wish to connect PatchWork to OMS application such as Max or Vision, you must use OpenShare. OpenShare is a small application that you launch before PatchWork and any OMS-compatible applications. OpenShare will remain in the background. Then you may use msConnect, in order to connect the input and output of PatchWork to whatever OMS application you like. MidiShare, OpenShare, msConnect, along with many other nice MIDI tools are provided to you by courtesy of the GRAME, with this PatchWork distribution.

Before using Midi with PatchWork, you must read : *Install MidiShare Readme*, and *MidiShare Tools ReadMe*

AppleScript

PatchWork is fully scriptable and recordable. See the "Pw-Script.pdf" documentation file in "PatchWork documentation:PatchWork Libraries Doc:"

Tutorials and documentations

It is strongly recommended to follow the PatchWork tutorial in order to learn the PatchWork basics. Open the file "PW-tutorial.pdf" in the folder "PatchWork documentation:PatchWork tutorial" and follow the path. In the same folder, you will find a folder named "40-Tutorials". It contains the 40 Patchwork documents that are commented in the tutorial.

If you own the first edition of the handbook, (April 1996), please note that the chapters "Hardware requirement", "Structure of folders", and "Launching PatchWork" are out of date, so you should rather refer to this Readme. Also, all references to PW 2.5.x should be replaced by PW 2.6.3. The new edition (Oct. 1996) has been corrected to match PW2.6.3ppc.

After reading the tutorial, you will find useful things in :

- PW-Introduction.pdf
- PW-Reference.pdf
- PW-Programming Guide.pdf
- PatchWork Libraries Doc:**

You will find some sample PatchWork documents in the folder Patchwork 2.6.3 ppc:documentation:sample patches.

All the tutorials are in Adobe pdf format.

Caution: some graphics are not perfectly legible at normal display size in these documents : if some graphics appear somewhat blurred, try a 110% zoom in order to get clean display.

Libraries

There are plenty of interesting libraries in the folder " Patchwork 2.6.3 ppc:user-libraries:".

To load a library, use the menu command `PWoper->Load Library`. Then choose a library folder and load the file that is suffixed by ".lib" (e.g. `esquisse.lib`). The modules belonging to the library will appear in the `Menu User`, at the right of the PatchWork menu bar. There is one exception: the `PWConstraints` library, once loaded, will appear in the `Music` menu.

Be careful with memory allocation when loading libraries as they may be quite heavy.

On-line doc and tutorial

Keyboard command 'd' when a module is selected inside a PatchWork window brings a documentation of the module. Command 't' brings a tutorial. Check inside the libraries folders : there are often folders containing sample documents (e.g. `user-library:profile:profile-patches:profile-examples:`).

What's new in PatchWork 2.6.3 ppc?

PatchWork 2.6.3 ppc has a smaller footprint in memory (5-7MB) than the previous beta version of PatchWork ppc. The theoretical minimum memory partition is now 8 MB, plus 4MB for the shared libraries.

All the shared libraries are installed in the image itself, so there is no more extensions to be installed in the system.

`PPCexceptionEnabler` (that used to crash with other Ircam software) is not required anymore.

`Abort` in the `PWoper` menu works.

`Lisp` in the `Apps` menu works, except if you have closed all the listeners.

PatchWork uses `MidiShare` for Midi input and output. It has now a very efficient and robust Midi play and record. You can play and record huge sequences while using the mac interface (even open and load files!) without noticing any desynchronization. You can connect PatchWork Midi input and output to `OMS` and `MidiShare` application, and you may even record in a PatchWork editor what's being played in another PatchWork editor! You may have real time `MidiPlay` even while `Lisp` is garbaging.

The Midi-files library is no longer of use in this version, as several Midifiles utilitie are now built-in. See below, `New PatchWork Modules`.

Module that have changed: Play-object

`Play-object` object `chan` &optional `approx` [`function`]

`play-object` plays a `note-object`, a `chord-object`, a `chord-line` object, a `measure-line` object (`rtm`) or a list of such, connected to it. Typing 's' while the box is selected stops the play.

`chan` if 0, plays with regard to the channel information contained in the object. if > 0, plays the object on the specified channel. Micro-interval channels are translated accordingly.

approx

The pitch approximation used. This can be the whole tone (approx = 1), semitone (approx = 2), quartertone (approx = 4, the default) or eighth-tone (approx = 8). Notes with microtonal accidentals are sent to different output channels according to the following mapping: chan (semitones), chan + 1 (eighth-tones), chan + 2 (quarter-tones), chan + 3 (three-eighths tones).

New PatchWork Modules

midi-record

midi-load

midi-save

midi-record



Syntaxe

```
patch-work::midi-record delta [function]
```

Midi sequences recorder.

Select the box and type 'p' to enter the record mode. Select the box and type 's' to stop the recording.

Output Menu

if 'chord' all the recorded notes will be collected into a chord. Output is a chord object.

if 'chord line' a sequence will be recorded. Notes falling in a time interval of 'delta' will be considered as chords. delta is expressed in 1/100secs. Output is a chord-line object.

Once the recording is done, the box may be evaluated as many time as required with different delta and output mode.

midi-load



Syntaxe

```
patch-work::midi-load delta [function]
```

MidiFile loader.

Once evaluated, issues a choose-file-dialog that lets you select a MidiFile. Any Midifile format, any number of

tracks is allowed. Tempo and tempo change is recognized. Channels are kept.

Notes falling in a time interval of 'delta' will be considered as chords. delta is expressed in 1/100secs.

Output is a chord-line object.

midi-save



Syntaxe

```
patch-work::midi-save objs [function]
```

MidiFile saver (saves Midifiles at format 0).

Once evaluated, issues a choose-file-dialog that lets you name a MidiFile. Generates a single-track MidiFile with tempo at Quarter note=60 and quarter note resolution = 480 (i.e. 480 pulses per quarter note). It handles correctly micro-intervals on different channels, so playing from PatchWork or from your favorite sequencer is the same.

Input may be any PatchWork object that could be played through play-object (chord, chord-line, measure-line (rtm) , or list of such)

Libraries updates

PWConstraints

Mikael Laurson has provided an impressive library called PWConstraints. A comprehensive manual is available on your CD-ROM.

PW-MIDI Manager

This library is no longer used with PW 2.6.3, since this release does not include Apple's MIDI manager, but GRAME's MidiShare.

PW-MIDI-files

This library is no longer used with PW 2.6.3. See the section "New PatchWork Modules" on page 5 .

Libraries current versions

Here are the latest version numbers of the libraries:

3Dim-disp	v. 1.0	(April 1994)
AAC	v. 1.0	(July 1995)
ALEA	v. 1.0	(October 1993)
Chant	v. 1.3	(October 1995)
Chaos	v. 1.0	(June 1994)
Csound-edit-sco	v. 1.2	(October 1995)
Esquisse	v. 1.0	(October 1994)
Functionals	v. 1.0	(February 1994)
Genutils	v. 1.0	(April 1996)
Instruments	v. 1.0	(October 1993)
Kant	v. 1.5	(September 1995)
Midi-file	v. 1.1	(September 1995) [Only with PW 2.5]
Midi-Manager	v. 1.0	(September 1995) [Only with PW 2.5]
Profile	v. 1.0	(October 1995)
PWConstraints	v. 1.0	(October 1996)
Repmus	v. 1.0	(April 1996)
Situation	v. 1.1	(February 1994)