SS

COLLABORATORS			
	TITLE : SS		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		May 29, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

SS

1.1 MANUAL

Wenn Sie OctaMED V6 kennen, können wir voraussetzen, dass sie die meisten
Befehle kennen und deshalb zum
OctaMED Soundstudio V1
Teil springen können!

Wichtiger Hinweis für CD Benutzer:

Sie werden das Wort PROGDIR in diesem Handbuch mehrmals finden, was das Laden und Speichern von Dateien betrifft. Auf CD können sie natürlich nicht speichern, so daß Sie die Pfade entsprechend auf die Festplatte, bzw. Diskette umlegen müssen, um Mods, Samples, etc speichern zu können.

1.2 Introduction

INTRODUCTION

Welcome to OctaMED Soundstudio V1, the main body of this manual covers all the features of OctaMED, however, a lot of new additions have been added and these are featured in the Soundstudio Features part of this manual, If you find something that you do not understand, check in that new section first of this manual.

The MED in OctaMED stands for Music EDitor, and that's exactly what OctaMED is: a versatile music composer and editor and although capable and sophisticated, OctaMED is also easy and interesting to learn. With the program you may compose music for games, demos, animations and presentations, or simply use the program as a stand-alone editor. OctaMED offers a multitude of features unseen on any other Amiga music package available to date.

In the early days of the Amiga, programmers needed a fast, convenient way to write music for games and demonstrations. A music program called SoundTracker was written in response to this need. It provided a perfect way to incorporate sound tracks into programs, and as a result, most Amiga games and virtually all demo programs had their music written with SoundTracker (or one of its many imitation programs, known as trackers).

Unfortunately, like many programmer's utilities, trackers required great patience. Although composing music was not especially difficult, they often crashed, seldom came with instructions, and refused to multi-task (one of the Amiga's major attributes). In spite of these problems, trackers became increasingly popular, a testament to their usefulness.

Recognizing the value of SoundTracker, Teijo Kinnunen began designing an

improved program. The first version, MED 1.12, appeared in Autumn 1989.
Teijo improved this version with new program code and, in April of 1990, he release MED 2.00. Perhaps the most important new feature was MIDI support. MED gained a reputation as an excellent piece of music software; virtually every Amiga publication of both sides of the Atlantic has praised MED.

In 1991, after being persuaded by Ray of RBF Software, Teijo released OctaMED, a great improvement on MED with its main new feature being the ability to play eight sounds at the same time, breaking out of the Amiga's four-sound limitation. With the release of OctaMED Professional V5, the user interface was completely redesigned to give the program a much more "friendly" appearance. The program thus moved into the 90's by becoming Kickstart 2 only, leaving many users with no choice but to upgrade.

When V6 was released in April 1995 OctaMED further modernized the program, with all gadgets in windows and flexibility of screen mode and font.
The much-requested Standard MIDI File support eas included, together with full

ARexx support

along with 16-bit and stereo samples etc.

This latest version contains some very new and special routines to allow mixing, recording samples to hd, up to 64 channels, extra new MIDI commands, and an awful lot more besides and because of these new special additions it was decided upon the name change at this time.
OctaMED Soundstudio is born!
Enjoy the world renowned and premier Amiga music package!

Credits

main

1.3 Credits

OctaMED Soundstudio itself

Teijo Kinnunen Programming, design and graphics
Ray Burt-Frost Suggestions, ideas, beta testing and taking care
of worldwide sales and distribution etc.
Ed Wiles, Andy Soar, Danny Amor, Dean Murray, Mat in Australia,
Tony Horgan and the many others who chipped in with bug reports and ideas.

Izrael Simila For Some OctaMED icons
Steve Hayes IFF sample loader routines (Fish disk number 64)
Ray/Danny/Ed for final Pre-release testing

Copyright and legal notices

OctaMED Soundstudio is NOT public domain or freely distributable.
It is a copyrighted program, available only directly from RBF Software

or via officially registered / contracted companies holding contracts
with RBF Software to retail the program on floppy only..

The CD-ROM version is also available directly from RBF Software but you
may also find it on sale in your local shop or via your favorite distributor.
If they haven't got it in stock, ask them to contact us.

You are not allowed to distribute/pass on OctaMED in any way.
However, as a registered purchaser you may make two backup copies for
retention in case of corruption.
These must be retained for your own, private use.

If you intend to use OctaMED Soundstudio (or any of versions 2 to 5) within commercial software or hardware, please note that as it is a fully copyrighted program, you are required by law to contact RBF Software for a licence. Any person or company found to be using this program for such a purpose without first having obtained a licence will be deemed to have broken the copyrighted usage of the program and solicitors will be instructed. The exception to this is the program titled OctaMEDPlayer, which may be treated as Public Domain. This will allow enthusiasts to pass out any songs they've composed to others, without breaking any copyright laws.

The OctaMEDPlayer and the player library may be freely used or distributed with non-commercial programs (PD or shareware etc.). You do not need permission from the owners to distribute them with your PD program(s). If you wish to make use of the source or library routines that are in the Programmers drawer then please read the file in that drawer.

Powerpacker.library is copyright (c) 1991-1992 Nico François.
Lh.library is copyright (c) Olaf Barthel.

OctaMED is provided "as is". Every effort has been made to keep OctaMED as bug free as possible, but if you lose anything due to any bugs, please note that by loading and launching this program you are agreeing to absolve the author or RBF Software from any liability, and you are agreeing not to pursue any claim for such losses be they deemed civil or criminal.

Anyone wishing to print and subsequently sell this OctaMED manual must
contact the copyright holder, R. Burt-Frost, for a licence before doing so.

If you have purchased the CD version then please note that you have
paid only for the program titled OctaMED Soundstudio and the MUG items.

The remaining files, mods etc, are provided freely and have been taken
from various PD sources, including items sent directly to us for inclusion
by their authors and also from files uploaded to the OctaMED User BBS.
The authors have declared them as PD/Freeware as long as no charge is
made for them and as we have only charged you for this new version of the
OctaMED Soundstudio program plus the two others items mentioned above,
we hope that you will accept that not all the mods will be of professional
standard, however, the composers have at least tried, so now it's your turn!!

To the authors, we send our greetings, and can only say, "don't forget".....
If Music Be The Food Of Life.........PLAY ON!! :)

Contact Addresses

main

1.4 Contact addresses

For any matters concerning distribution, sales, copyright, licencing, or such, you should contact the world copyright owners RBF Software:

Address: 169 Dale Valley Road, Hollybrook, SOUTHAMPTON S016 6QX, England
Phone: +44 (0)1703 785680 (Ansa/Fax)

Note that this phone is unmanned at most times, so leave a short message and we will get back to you as soon as possible, or, if you have a fax, then please fax us with your enquiry.
You can also E-mail us on:rbfsoft@octamed.co.uk

OR

rbfsoftcix.compulink.co.uk

For matters concerning the program itself, e.g. suggestions, ideas for any future Amiga versions, (depending on if the Power version arrives), bug reports, you can write to the author Teijo Kinnunen:

Address: Oksantie 19, FIN-86300 OULAINEN, FINLAND
E-mail: kinnunenstekt.oulu.fi

Main Features

main

1.5 Main Features

This really is only an introduction to OctaMED's main features: the manual cannot cover all the program's functions. However, if you discover anything in OctaMED that you're unsure how to use, simply press the Help key for full instructions.

On loading OctaMED, three windows should appear. Main Control provides access to some important functions, and Information displays useful information. The middle window is called the Tracker editor; this is where you create songs.

The best way to learn about creating an OctaMED song is to load in a demo

song and examine it. To load a song, hold down the right mouse button and
select Project menu -> Open (on the left-hand side of the title bar).
A file requester appears; you'll encounter one often when using OctaMED.

Now, besides all the other drawers you will see listed, sitting there
all on it's own, is a module called "MISERY", double-click on it's name
and it will load into the program.

To listen to the loaded song, find the playing buttons on the left-hand side
of the Main Control window. Click on the Play button beside Song.
When you've had enough, click STOP or press the space bar.
The Cont button continues the song from where it left off.

Now let's examine the components which make up the song. Note: Shift-<right>
(for example) means hold down a Shift key and press the right arrow key,
by looking at

The Instruments

Creating a Song

Editing a Song

main

1.6 The Instruments

The current instrument's details are displayed at the top of the Main
Control window. 01 is its number, Major Strings its name and 4911 its size
in bytes. Select the next instrument (number 02) by pressing Shift-<right>.
Repeatedly press Shift-<right> to view all the instruments in the song.

Notice that not all instrument 'slots' (available spaces) are used
by a song, also notice that the numbering system used is
unorthodox: 01 - 09, 0A - 0V, 10 - 19, 1A - 1V. 1V is the final
ins slot, the 63rd.

OctaMED can use four different 'types' of instrument, and easily the most
commonly used type is the sample.
Samples are sounds from real-life, often created by feeding the sound into
the computer from a sound source (e.g. tape recorder).
All the instruments used in this song are samples.

Repeatedly press Shift-<left> until the instrument number becomes 01. Now
let's try playing the instruments; this is done using the Amiga's keyboard.
OctaMED arranges the keyboard into about 2½ piano-like octaves.

Piano keyboard layout: the notes of one octave

```
| |  | |  |  | |  | |  | |  | |
| |  | |  |  | |  | |  | |  | |
| |C#| |D#|  | |F#| |G#| |A#| |   <== Black keys (sharps)
| +--+ +--+  | +--+ +--+ +--+ |
                |   |    |   |   |    |   |   |
                | C | D  | E | F | G  | A | B |   <== White keys (naturals)
                +---+----+---+---+----+---+---+


                         Amiga keyboard layout

    Lower octave   Middle octave   Upper half-octave

                 |  2 3   5 6 7  |  9 0   = \     <== Black keys (sharps)
                 | Q W E R T Y U | I O P [ ]      <== White keys (naturals)
                 |               |
    S D   G H J  |  L ;          |                <== Black keys (sharps)
   Z X C V B N M | , . /         |                <== White keys (naturals)
```

Try playing notes with the keys listed above. For example, press the following keys in turn to play a C major scale: Z X C V B N M Q W E R T Y U I O P [ ]. Use Shift-<left> and Shift-<right> to change the instrument the notes are played on.
A song is made up of several small pieces of music called blocks, so let us look at

Using The Blocks

Creating The Song

Editing The Song

main

1.7 The blocks

Blocks consist of many lines of information, and are displayed in the
Tracker editor (the middle window).
Only one block can be displayed at a time.

All the blocks in this song contain 64 lines and 4 columns (or tracks). The lines are numbered 000 - 063 on the far left, and the tracks are numbered in small blue squares at the top of the window. Try using the <up> and <down> keys to scroll through the block. You can also do this by dragging the grey or white scroll bar at the far right of the Tracker editor.

In the Information window, find the block status display (to the right of the B button). It will display something like: 6/25: Synth (part 2)

In this example, block number 6 is currently being displayed in the Tracker editor; that is, it's the current block. The current block's name is Synth (part 2), and block number 25 is the last block in the song. Note that the first block's number is 0 (zero) rather than 1.

Watching the changes to the block status display and the Tracker editor, try using Shift-<up> and Shift-<down> to select a new current block.
Left Alt - <up> / <down> selects the first / last block.

To listen to the current block, find the playing buttons on the left-hand side of the Main Control window. Click on the Play button beside Block. Use STOP or the space bar to stop the block, and Cont to continue the block.
When the block has played through, it will start again. Stop the playing when you've finished.
So just what do all these letters, numbers and dashes displayed in the Tracker editor mean?
Lets take a look at

The block contents

Using The Blocks

Creating The Song

Editing The Song

main

1.8 The block contents

The Amiga plays music using four sound channels, each of which can play
one sample at a time.
The four tracks (columns) in the Tracker editor correspond to these sound channels.

Music is composed by entering notes on to these tracks. A typical note might be C-2 10C32. Here's how this note is made up:

```
  octave    player command type
  number      /|
       |     / |
   C - 2   1 0 C 3 2
   |       |     \ |
note    instr.    \|
name    number   player command level

                 3
```

So note C is played on instrument number 01 using octave number 2. The note also has a player command (0C32), which means 'set the note's volume to level 32'. Octave numbers and player commands will be explained later, but first, let us look at

The playing sequence

Using The Blocks

Add new blocks

Creating The Song

Editing The Song

main

1.9 The playing sequence

To create a song, the blocks are linked together. This involves creating a list of blocks in the order in which they should be played. This list is called the playing sequence.

Look carefully at the Information window, and notice that there are two buttons that seem to be called Sq. Well, the left-hand one is actually Sg. Click on the other one to open the Playing Sequence window.

In this window you'll find a list of block numbers and their names. On the left are two columns of numbers. The left-hand column contains the playing sequence line numbers, the right-hand column contains the block numbers on each line. Although the line numbers are in ascending order, the block numbers can be in any order and the same number can feature more than once in the list.

When the song is played (by clicking Song Play in the Main Control window), the song starts playing from playing sequence line 001. Each block in the sequence is played in turn, with the block currently being played highlighted by a blue bar.

Close the window by clicking either its close gadget (far top left) or its Exit button (bottom).

OK. We've had a look at the basic structure of a song, now let's examine how to

Create a Song

The block contents

Using The Blocks

Add new blocks

Editing The Song

main

1.10 Create a Song

Firstly, clear the demo song by selecting Project menu -> New then clicking Clear All in the window that appears.

By default OctaMED has no instruments in memory; they must be loaded in. You can either select Instr menu -> Load Instrument(s), or click the small GetFile button sandwiched between the instrument number and name in the Main Control window.

Try loading in some samples (if you don't have any, most PD libraries will be happy to help). Remember to select the instrument number first using Shift-<left> and Shift-<right>. Use the Volumes button in the file requester to select the correct disk. You can select more than one instrument by holding down Shift and clicking on several files (then click Ok); the instruments are then loaded into successive instrument slots.

Firstly, select the instrument that you wish the note to be played on. Use Shift-<left> and Shift-<right> as usual. Next, position the cursor where you wish the note to be in the Tracker editor. Use <up>, <down>, Alt-<left> and Alt-<right> for this.

Now switch edit mode on by clicking on the Edit check box (Main Control window), then play the note using the keyboard (see the diagram in

The Instruments
for a reminder.
The note name, octave number and instrument number is entered.

Just what is this octave number? Well, OctaMED arranges the Amiga's keyboard into 2½ octaves.
However, OctaMED's instruments can play using more octaves than this, depending on the instrument's type.
Samples, for example, can play using 3 octaves, numbered 1 - 3
(this is the octave number).

So how do you play the extra octaves? Well, by default the keyboard's lower octave (Z - M) plays using octave number 1, the middle octave (Q - U) using number 2, and the upper half-octave (I -]) using about half of 3. This can be changed by using function keys F1 - F5.

Find the Oct cycle gadget in the Main Control window. The 12 refers to the lower octave and middle octave's current octave numbers, 1 and 2 respectively. Now press the F2 key. The Oct gadget is now 23. So the lower octave's number is 2 and the middle octave's 3. (The upper half-octave uses about half of octave 4, but since samples can only play using 3 octaves you can't use octave 4 with samples).

Confused? Try entering a few notes, using F1 to select octaves 1 and 2 and F2 to select 2 and 3, and examine the octave number of each note.
For example, how about a bass drum instrument on track number 0 every fourth line (000, 004, 008...), and maybe a hi-hat on track 1 every second line?, stick with something really simple like that to start with, and it shouldn't be long before entering notes becomes second nature.

Remember to use the Block Play button in the Main Control window to play the block through. Remove notes (that is, enter a blank note: --- 0) by positioning the cursor over the offending note and pressing the Del key.
Now let us see how to

Add new blocks

The block contents

Using The Blocks

Add new blocks

Creating a Song

Editing The Song

main

1.11 Add new blocks

How do we add new blocks and maybe change their length as well?

By default there is just one block in the song, number 0. To add a new block, select Block menu -> New -> Append. Look at the block status display in the Information window and notice that the last block in the song is now number 1. Select this block by pressing Shift-<down>, and return to the previous block using Shift-<up>.

Usually blocks are 64 lines long. (Note for musicians: this allows for 4 measures of 4/4 time; each line is one sixteenth-note). You can change this length if you so desire by selecting Block menu -> Set Properties. Click inside the Length box in the window that appears, delete the 64 using the Del or Backspace keys, and type in the new length and press Return (maximum length is 3200 lines).

In the Block Properties window you can also name a block by typing its name into the Name box. (Its name appears in the block status display). Using the Tracks slider you can change the number of tracks (maximum is 64), but this is only useful when using 5 - 8 channel mode or MIDI (explained later).

Open the Block List window by selecting Block menu -> Block List. This window provides a list of all blocks and their names. The blue bar highlights the block currently shown in the Tracker editor; select another block by clicking on its name or number in the list. The Append New button is the same as Block menu -> New -> Append; Insert New inserts a new block at the current block position (same as Block menu -> New -> Insert).

The Block List can also be opened by clicking the small B button in the Information window. Holding down a Shift key while clicking this button opens Block Properties.
So, let’s see how to

Link the blocks together

Adding new blocks

The block contents

Add new blocks

Using The Blocks

Creating a Song

Editing The Song

main

1.12 Link the blocks together

To create a song blocks are linked together by creating a list of ←↩ blocks
in the order they should be played.
This list is called the playing sequence, you can open the Playing Sequence
window by selecting Song menu -> Playing Sequence.

There is always at least one line in the list, and by default it’s block 000
Change this number using the two arrows just below the list (note that
the number can’t be increased beyond the number of the last block in the
song). Add a new line by using the Insert or Append buttons, in the same way
as for blocks. In the Block List, the Ins to Seq and App to Seq buttons
insert and append the block currently shown in the Tracker editor to the
playing sequence.

The display box just below the list shows the current playing sequence line
number and the total number of lines in the list. This is duplicated in the
Information window. When Follow is switched on, the current playing sequence
line number is kept as close to the middle of the list as possible during
play (try clicking Play Song with Follow on to see this effect).

Now let us see how to

Save the Song

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

The block contents

Using The Blocks

main

1.13 Save the Song

Select Project menu -> Save. If there is no filename in the box at ←↩ the top
of the Save Options window, a file requester appears to select a name.
Switching Create Icon off prevents OctaMED saving an icon file with the
song (for use with Workbench and especialy if you save an executable mod).

Clicking repeatedly on the No Compression cycle gadget selects one of several forms of compression (or none at all), reducing the file size. Calculate Size calculates the entire song’s size using the current settings (use the Packed button to calculate the song’s size when compressed).

Click Save to save the song.

Right, have you got all that?

You have?, ok then, let us now move on and see the

Player Commands

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

The block contents

Using The Blocks

Saving the Song

main

1.14 Player Commands

So far we’ve ignored the 0000 appearing to the right of Tracker editor note names. Well, the four digits are collectively known as the command digits. They allow you to add special effects to the notes, and to ask OctaMED to do certain things. These effects and actions are called player commands.

The command digits are divided into two pairs:
The command type is the sort of effect or action, and the command level allows you to control the effect or action’s intensity, depth, speed etc. (depending on the sort of command).

For example, if you were to see a note with command type 0C and level 32, 0C means ’set volume’, so the player command sets the note’s volume level to 32. A full list of player commands is given in the Appendix section A.

Player commands are entered one digit at a time, simply by moving the cursor over to each digit (using the <left> and <right> keys) and pressing the correct key. You’ll find that turning Advance Cursor Right on will help (select Settings menu -> Keyboard Options and click Right - far top left). Also, to enter command types such as 19, position the cursor over the right-hand command type digit and press Alt-9 (the left-hand digit is set to 1).

Both the command type and command level are in hexadecimal. If you’re unfamiliar with hexadecimal, please read Appendix section D.

A rough understanding of hexadecimal is important for successful use of player commands.
Also important is an understanding of

Slide creation

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

main

1.15 Slide creation

Slides can be created automatically, without having to ↩ painstakingly enter
each individual command in the slide. This is done using the Pitch, Volume and Generic Slide items in the Edit menu.

To use the Pitch Slide function, firstly enter the notes that the pitch is to slide from and to. (The notes must be on the same track). Now position the cursor where the pitch slide is to begin; this will be somewhere between the two notes. Finally, select Edit menu -> Pitch Slide -> Type 1 or Type 2 (try out both: type 1 uses command type 03, type 2 uses types 01 and 02).

For Volume Slide, you need to enter two 'set volume' (command type 0C) commands in the same track: one to set the starting volume level, one to set the ending level. Then position the cursor anywhere between the two commands and select Edit menu -> Volume Slide. Generic Slide works in the same way except that any command type (not just 0C) can be used.

Of course, the previous sections are only a very simplified guide to song creation, in particular, OctaMED provides a number of editing features for use in the Tracker editor that can make entering notes much easier.

Now let us move on to actualy

Editing The Song

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

main

1.16 Editing The Song

The commands Cut / Copy / Paste / Swap

You may recognize the terms cut, copy and paste from wordprocessors. Well, in OctaMED they are exactly the same: Cut removes notes and stores them in a copy buffer, Copy stores notes in a copy buffer without removing them, and Paste transfers the copy buffer to the Tracker editor, overwriting any existing notes.

Items in the Block and Track menus allow you to cut, copy or paste whole blocks or whole tracks respectively. For example, to place a copy of track 0 in track 3: move to track 0 (using Alt-<left> and Alt-<right>), and select Track menu -> Copy. Now move to track 3, and select Track menu -> Paste. The Swap items swap the current block or track's contents with the copy buffer's contents.

You can also cut, copy and paste a rectangular area of notes, known as a range. A range can be as small as a single note or may include an entire block. Mark a range by moving the mouse over notes in the Tracker editor while holding down the left mouse button. The range appears in white. Moving the mouse beyond the limits of the Tracker editor continues the range in the relevant direction (for example, try moving the mouse to the bottom of the screen while holding the left mouse button).

The Edit menu contains Cut, Copy and Paste Range items. Erase Range changes all the notes in the range to blank notes (--- 00000). If you mark a range by mistake, press the left mouse button again to cancel it.

Now we will take a look at the

Transpose Window

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

main

1.17 Transpose Window

Select Edit menu -> Transpose. This window contains functions that alter the notes in a particular area of the current project.

The gadgets under Affect should be selected before choosing a transpose function. They select which area of the song the transposition should affect, and the notes played by which instruments. The pitch-changing functions are Octave Up / Down and Halfstep Up / Down:
these should be self-explanatory.

The Change Notes functions act on the Source and Destination notes. Select these notes by holding down the left mouse button on each note box in turn and pressing the appropriate key (for example, the I key selects note C-3). The Change button changes all notes in the Affect area (see above) from the source to the destination note. Swap swaps the source and destination notes.

The Change Instrument functions act on the Source and Destination instruments. Select these instruments by choosing each required instrument (using Shift-<left> and Shift-<right>) and clicking the Source and Destination buttons. The Change button changes notes played by the source instrument to the destination instrument (in the Affect area). Swap swaps the source and destination instrument numbers of notes played by either. Delete deletes notes played by the source instrument.

Ok, lets now look at how to use

Track Selection

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

```
                 Saving the Song

                 Player Commands

                 Slide creation

                 Transpose Window

                 main
```

1.18 Track Selection

```
                Look at the top of the Tracker editor; the blue boxes are the  ←↩
                   track
selectors. They are blue when on, grey when off.
Try clicking on a few to turn them off and on.

You'll notice that there are two types: one displaying the track numbers,
one displaying S. The track number type control whether each track is played
or not played. For example, load a song and click Song Play (Main Control).
Now keeping the song playing, try turning the track numbers on and off. Do
you notice the notes in the related tracks being turned off and back on?
This can be very useful while editing blocks.

The S stands for Selected, and this type control whether each track is
selected or not selected. They are used in editing functions such as
Edit menu -> Paste To Selected Tracks and in the Transpose window (Affect
Selected Tracks).

At this stage it would be a good idea to looK at the
                 Programmable Keys

                 Creating a Song

                 Editing The Song

                 Linking blocks

                 Adding new blocks

                 Block contents

                 Using The Blocks

                 Saving the Song

                 Player Commands

                 Slide creation

                 Transpose Window

                 Track Selection
```

main

1.19 Programmable Keys

(progkeys)

The progkeys allow music to be entered much more easily and quickly.
You can assign notes or groups of notes, including player commands,
to 10 different keys.
The assigned notes can then be inserted anywhere in the Tracker editor
by holding down Shift and pressing key 0 - 9.

Select Settings menu -> Programmable Keys. The cycle gadget in this window selects whether you wish to edit the normal Shift- 0 - 9 definitions (default) or the Right Alt definition. The Right Alt definition consists of the command digits only, and if you enter notes with the Right Alt key held in the Tracker editor, these command digits will be inserted with the notes.

Select which progkey you wish to edit (0 - 9) using the slider. The box containing --- 00000 is the definition of the selected progkey. Edit the progkey one digit at a time, by holding the left mouse button on each digit and entering a new note or digit using the keyboard.

To make a digit transparent (shown by an x), press Return while holding the left mouse button. A digit being transparent means that when the progkey is entered in the Tracker editor, the corresponding existing digit will remain unchanged. For example, the definition xxxxx0000 would set all command digits to zero while leaving the note and instrument number unchanged.

The Pick gadgets copy either the note under the cursor, the currently marked range, or the copy buffer to the current definition.
Whilst you are here, take a look at

The Numeric Keypad

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

main

1.20 The Numeric Keypad

The numeric keypad is the small collection of 18 keys on the far right of your Amiga’s keyboard (excluding A600 owners). Without alteration (using Settings menu -> Keyboard Shortcuts), the keypad is layed out as follows:

Keys 1 - 9	Select instruments 1 - 9
Key 0	Select instrument 10
Key /	Select last used instrument
Key .	Change the first instrument digit (e.g. 05 -> 15, 15 -> 05)
Key + / -	Next / previous instrument
Key (/)	Decrease / increase current instrument’s volume by one
Key *	Pick instrument number nearest the cursor
Key Enter	After pressing Enter, press key A - V to select the corresponding instrument

Some other editing functions in brief

Space (Main Control window): when on, a set number of lines are skipped when a note is entered. Select Settings menu -> Keyboard Options and adjust the Space Value slider to set the number of lines (default = 2).

Expand/Shrink (Block menu): Expand creates empty lines between each note, and Shrink removes lines. If the Factor box in the window contained 3, for example, clicking Expand would insert two empty lines between each note (trebling the block length), and clicking Shrink would remove every second and third line (thirding the block length). It has the effect of slowing or quickening the music. Should become clear with some experimenting.

Highlight Options (Block menu): Here you can highlight the current block’s lines in a particular order. This can mark measures and beats for you. Try clicking the 4 gadget to highlight every fourth line: in a normal 64-line block, this highlights every beat.

So now that you are confident with all you have read so far, WHAT?
your not!, then read it all again!, we will look at

The Sample Editor

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

1.21 The Sample Editor

Making sure a sample is loaded into the current instrument slot, select
Display menu -> Sample Editor (or click Edit in the Main Control window).
The squiggly black line in this window is the sample’s waveform,
which is a graphic representation of the sample.

The sample editor is used to change the whole or part of the waveform, which
can involve copying sections of waveform to other places in the sample,
changing the volume or pitch of the waveform, creating echo effects, and
much more.

The white line across the center of the waveform marks zero volume.
The further away from the white line a point on the waveform is,
the louder the point’s volume.

(let us take a look at a diagram...)

```
        Volumes of points on the waveform display

    Display: 0032098 Buffsize: 32098 R Start: 0 R End: 0
   +-----------------------------------------------------+ <- full volume
   |                                                     |
   |             TOP HALF OF WAVEFORM DISPLAY            | <- half volume
   |                                                     |
   +---------------------white-line----------------------+ <- zero volume
   |                                                     |
   |           BOTTOM HALF OF WAVEFORM DISPLAY           | <- half volume
   |                                                     |
   +-----------------------------------------------------+ <- full volume
```

The number to the right of Buffsize (above the waveform in the middle) is the size of the current sample in bytes. (In the diagram above, the sample's size is 32098 bytes). Type a new number into this box to change the sample's length.
You can use the sample Zoom In and Out to get
Magnification
Lets see how to use the
Sample Digitizing Window

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

main

1.22 Magnification

Try using the Zoom In and Zoom Out buttons to magnify and reduce the display. The black horizontal scroll bar just below the display shows the section of sample currently being displayed: drag it or use the <left> and <right> keys to change the current sample section. The Display box (top left) shows how many bytes of sample are currently displayed.

You can also magnify and reduce the display using the small vertical zoom scroll bar (far right). The higher the scroll bar, the greater the magnification. Click Play Display to play the current display at the pitch shown just below the display on the right (note C-2 by default). Change this pitch by holding down the left mouse button on the pitch box and pressing a new note. When you've finished, click Show All to display the whole sample.

Take a look at the

Sample Range Operations

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Sample Digitizing

Magnification

main

1.23 Sample Range Operations

Just as in the Tracker editor, you can mark a range by moving the ↩ mouse over the waveform while holding the left mouse button. Hold a Shift key while doing this to adjust an existing range. The byte positions of the start and end of the range are shown by Range Start and Range End (top right). Click Range Display to mark a range over the whole display.

Eight range-affecting buttons are situated near the bottom of the window. Show magnifies the range to fill the whole display. Play plays the range at the current pitch, cut and Erase both remove the portion marked by the range: Cut also copies the range to a copy buffer. Clear clears the range, Copy copies the range to a copy buffer, and Paste inserts this copy buffer at the start of the range. Reverse turns the range backwards.

Some other range-affecting operations are hidden in the Edit menu.
Paste (Overwrite) is like Paste except it overwrites the existing sample contents rather than inserting.
Invert turns the range upside-down, Chop removes the whole sample except for the range.
In the Tools menu, Calculate Range Time works out the time taken for the range to play at the current pitch.

At this stage, why not take a look at some of the
Special Effects
or how to us the
Sample Digitizing Window

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

main

1.24 Special Effects

These are all found in the Effects menu, and with the exception of Change Pitch and Mix, all affect the current range. Here are the most important effects in brief:

Change Volume changes the range’s volume. The sliders select the starting and ending volume change, and both are percentages of the original volume. For example, setting Start to 200% and End to 50% fades downwards from double the original volume to half the original volume. Click CHANGE VOLUME to change the volume, or choose from one of four presets (Fade In, Fade Out, Halve, Double). Max increases the volume to the maximum possible without introducing distortion. (Distortion occurs when the maximum volume limits are exceeded).

Mix mixes the current sample (source) with the copy buffer (destination), storing the result in the current sample. The sliders control the volumes of the two waveforms to be mixed (50% is ’normal’ volume).

Filter/Boost filters (reduces noise) and boosts (brightens) the range. The Averaging slider controls the strength, and try cranking Distance right up for some weird effects.

Echo allows interesting echo effects to be produced. Before echoing, you usually need to add some extra space to the end of the sample: select Tools menu -> Add Workspace to do this. Mark a range over both the waveform to be echoed and the blank space that the echo is to affect. Echo Rate is the distance, in bytes, between two echoes. Number of Echoes sets the total number of echoes, usually quite low (1 - 10).
Experimentation is the key to this feature.

Create Chord creates chords of two to four notes from the current sample. Specify the chord’s notes by pressing keys while holding the left mouse button on the pitch boxes (you can also use the eight preset buttons or the four cycle gadgets). Click Play Chord for a preview, and Create Chord to create the chord and store it in the current sample.
You can, of course also include

Sample Loops

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

main

1.25 Sample Loops

A sample loop means that the notes you play with the sample will ↩ be
sustained until stopped. This is due to a particular part of the sample
being continually repeated (or 'looped').

Try clicking the Loop check box (bottom left). The two loop pointers mark
the looped section. One way to adjust them is to drag their small black
triangles across the waveform. You can also use the Loop Point gadgets
(bottom of window): use the cycle gadget to choose whether to affect the
start or end loop pointer, then use < and > to move the pointer in steps of
two bytes, or <0 and 0> to move the pointer until a zero is found. The Loop
menu also helps loop positioning: see §12.2.5 for an explanation.

Decent-sounding loops often start and end at the same value, which can often
be zero (i.e. no volume). So <0 and 0> are useful in finding good loop
points. Magnification (see §5.1) allows far more accurate loop positioning,
and good loops are often found by looping any repeating waveform shapes in
the sample (called envelopes).

There are also some

Other Instrument Features

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

main

1.26 Other Instrument Features

Instrument Parameters

Select Instr menu -> Set Parameters or click Inst Params in the Main Control window. With this important window you may alter the characteristics of the current instrument.

From the top down, the slider and arrows select the current instrument in the same way as Shift-<left> / <right>. 1st / Last / L.U. select instrument 01 / 1V / the last used instrument. Flush removes the current instrument from memory.

Set an instrument’s default pitch by holding the left mouse button on the box and pressing the appropriate key. Pressing key F then inserts the instrument at this pitch into the Tracker editor.
Useful for percussion instruments whose pitch is often constant.
Repeat and RepLen are the loop start and loop length in bytes..

Transpose raises or lowers the current instrument’s pitch in halfsteps. Finetune allows you to tune instruments in small steps. Vol sets the instrument’s default (usual) volume: 64 is maximum, 0 is silent.

Now it’s time to take a look at

The Sample List

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

Inst Features

main

1.27 The Sample List

Select Display menu -> Sample List Editor. The sample list is a list of all the instruments you have in your collection (not just samples in fact), and is an easy way of organizing and loading instruments.

Notice that the Sample List Editor is split into two lists - a list of your sample directories, and a list of the samples contained in those directories. There may in fact be no directories shown: this is because you haven't yet told OctaMED in which directories your samples are contained.

To add more directories, click Add Dir and select the appropriate directory in the requester that appears. Repeat for all the directories you wish to add. OctaMED will load the names of all files in the directories; if some of these files aren't instruments, click on them in the left-hand list and click Remove.

To load an instrument into the current slot, click on the instrument name then click Load Inst. Alternatively, close the Sample List Editor and click SList in the Main Control window. This is a cut-down sample list editor used specifically for loading instruments. Select the appropriate directory, then simply click on the instrument name to load it. Select the current instrument using the four Prev / Next Inst / Free buttons (or Shift-<left> and Shift-<right> as usual).

In the Save Options window, the Save Instruments check box chooses whether the song should be saved together with its instruments. If you switch it off, only the instruments names are saved with the song; when the song is reloaded, OctaMED uses your sample list to locate and load each required instrument from your sample disks. This saves a lot of disk space.

You can save an instrument to disk using Instr menu -> Save Instrument.

And for all you MIDI freaks, let’s look at
The MIDI Features

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

main

1.28 The MIDI Features

OctaMED can output notes and some player commands to external MIDI devices
(such as MIDI synthesizers) by way of a MIDI interface which you should
connect to your Amiga's serial port.

Note: please refer to your MIDI device's manual if you come across terms in
this section that you aren't sure of.
To glean even more information, press Help then select Menus and MIDI.

A MIDI instrument in OctaMED is not strictly an instrument, but simply a few settings which result in the notes played with that instrument being sent through a MIDI interface. Therefore, MIDI instruments are not loaded into memory like other sounds.

MIDI instruments can also use the full 10½-octave range. Use the Oct cycle gadget (Main Control window) or the F1 - F5 keys to select these. F1 and F5 cycle through three different octaves if you press them repeatedly.

Before using a MIDI instrument, you need to change some settings in the Instrument Parameters window to the required values. MIDICh sets the MIDI channel for the instrument (1 - 16). For example, setting it to 5 means this instrument's notes are sent through MIDI channel 5. Preset sets the instrument's preset number. When set to zero, OctaMED uses your keyboard's default preset for the instrument's MIDI channel.

In order to hear a MIDI instrument, you must also set its default volume to above zero. MIDI instruments can also use all 64 possible tracks (use the Block Properties window). In addition, you may mix Amiga samples and MIDI instruments on the first four tracks.

The MIDI menu contains more MIDI-specific functions and settings. MIDI Active activates MIDI when selected. Input Active, when selected, allows the entering of notes into the song using your MIDI device. Edit mode (Main Control window) and MIDI Active must be on.

Read Key-Up's, when switched on, records key-up events into the Tracker editor (as 0FFF player commands). Read Volume: If you have a touch-sensitive keyboard, the volume will be entered as a 'set volume' player command (0C) when notes are input.

Reset Pitch/Presets resets pitchbenders, modulation wheels and presets on all channels. (OctaMED sends preset change messages for all MIDI channels).

There are a few
Other Miscellaneous Features
that you should look at.

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

The MIDI Features

main

1.29 Other Miscellaneous Features

The Tempo window

Select Display menu -> Tempo Window. This window is used to set the playing speed (tempo). For basic use, ignore all gadgets except the Tempo slider. The higher the Tempo slider value, the quicker the speed. For more advanced use, read on.

There are two tempo modes: SPD (the default) and BPM. Let’s look at SPD mode first. The Tempo and TPL sliders are used in this mode. Each slider alters the tempo in a different way, and to fully understand the difference I'll need to explain a few technical aspects. Don't worry, it's all quite simple really!

To keep 'in time', OctaMED responds to regular ticks that come from one of the Amiga's clocks. The tempo depends on two factors relating to these ticks:

(1) The time between each tick (Tempo slider)
(2) The number of ticks per Tracker editor line (TPL slider)

So each Tracker editor line is divided into ticks. The TPL slider controls the number of ticks per line. With TPL set to 3, two typical consecutive lines look like this:

```
                                   Line no.  Tick no.   Action
                               /     000        1       Play note C-3
                              /      000        2
 -000 C-3 10000 --\ ZOOMED --\       000        3
 -001 D-2 10000 --/ ZOOMED --/       001        1       Play note D-2
                              \      001        2
                               \     001        3
```

So notes are played on the first tick of each line. The other ticks are used for effects such as pitch sliding (although using certain player commands, notes can also be played on the other ticks).

The Tempo slider controls the length of time between each tick. The higher the value, the shorter the length of time therefore the faster the tempo. For normal 4 channel or MIDI songs, Tempo slider values 11 - 240 should be used (ignore values 1 - 10). The TPL slider can be values 1 - 32.
The higher the value, the slower the tempo. By default the sliders are set to 33 and 6 respectively.

In BPM mode, the Tempo slider controls the number of beats per minute (just like a metronome). For example, a value of 60 is one beat per second. The LPB slider controls the number of lines per beat. For example, 8 (the default) means eight lines are considered as one beat. In this mode, it's best to leave the TPL slider at 6, otherwise the timing won't be accurate.

If you are interested in the
Channel Modes 5-8
then read this.

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

1.30 Channel Modes 5-8

When composing using OctaMED, you may find that the Amiga's ↩ ability to play a maximum of four samples at once (using its four sound channels) is a bit limiting. Luckily, OctaMED allows you to add up to four extra tracks, extending the maximum number of simultaneous samples to five, six, seven or eight; and this feature is called 5 - 8 channel mode.

Firstly, increase the number of tracks to 5, 6, 7 or 8, using the Block Properties window (see §2.3). Next, select Song menu -> Set Options. On the left of this window, click on the appropriate channel mode button. Click Yes and Halve in the requesters, then close the window. Voilà!

Unfortunately, you'll notice some side effects pretty quickly. The volume of all samples halves to minimize distortion. There's a different method of setting the tempo (see §8.1): in SPD mode, you can only use levels 1 - 10. And the quality decreases, but try selecting High Quality Mode in the Song

Options window (if you own an A500 or A600, this will only work in 5 or 6 channel mode).

For more details, press on-line Help and select Special Purpose Topics.

Using the

Multi-Modules
project

main

1.31 Multi-Modules

Multi-modules are projects containing more than one song. Each song shares the same set of instruments. This allows you, for example, to compose different songs for each level of a game, without having to load or keep the same instrument in memory more than once.

Multi-modules are also very useful when editing just one song, as you can treat one song in the multi-module as 'workspace', much like Deluxe Paint's spare screen option. You may design and compose in one song, then cut and paste to another.

Now lets take a look at

Adding and Deleting Songs

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

The MIDI Features

Other Features

Channel Modes 5-8

Multi-Modules

main

1.32 Adding and Deleting Songs

Find the Sg 01/01 < > gadgets on the left of the Information window.

Sg opens the Song Selector window, which allows you to add and delete songs in a multi-module, and to choose the current song.

The two numbers (currently both 01) represent the current song and the total number of songs in the multi-module. < selects the previous song. If clicked while holding a Shift key, it deletes the last song in memory after displaying a requester. > selects the next song. If the current song is the last song, it adds a new song after displaying a requester.

Clicking < and > while holding Shift is the same as selecting Delete Last and Add New respectively from the Song menu.

You can also

Remap Instruments

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

1.33 Remap Instruments

```
                OctaMED can automatically remap instruments while loading. This  ↩
                   feature can
save you a lot of work. If you have one or more songs already in memory,
and you load an additional song, a Remap instruments? requester will
appear.

If you click No: all previous instruments in the entire multi-module will be
lost and the instruments of the new song will be loaded. This could be used,
for example, to use the new, different instruments with the other songs in
the module.

But if you click Yes:

   (1) The new song's instruments will be loaded into empty places, so that
       instruments already in memory are preserved;

   (2) The new song's notes' instrument numbers will be redirected
       (remapped) so that they point to the new instrument positions.

By clicking Yes, several sets of instruments used in several songs can be
```

contributed to one large overall set of instruments. This is the option you’ll most often (if not always) choose.

In the Save Options window: the right-hand cycle gadget selects whether to save the whole multi-module (default) or only the current song.

Add/Del Songs

Remap Instruments

main

1.34 Some Other Small Features

Select Song menu -> Set Volumes. This window full of sliders is used to set each track's volume and an overall (master) volume. The 'relative' part in the window's title bar refers to the relativity between the track and master volumes. For example, setting both master and a track volume to half (32) will produce an actual track volume of a quarter (16).

Select Song menu -> Set Options. Set the current song's name (different for each song in a multi-module) by typing it into the Name text box. The name will appear on the screen's title bar.

Many different settings (preferences) can be saved using Settings menu -> Save Settings. For a full list of saved settings, press Help and select Special Purpose Topics (then click The Settings File).

In your country, if the note between A# and C is H instead of B (for example Germany and Finland), select Settings menu -> Miscellaneous, then click H -> B in the window and save settings.

A little bit about

Installation

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

The MIDI Features

Other Features

Channel Modes 5-8

Multi-Modules

Add/Del Songs

Remap Instruments

main

1.35 Installation

I know installation is boring, so if you have purchased the floppy version with your name/address encoded into it, simply click on the install icon and away you go! ;)

IMPORTANT NOTE TO PURCHASERS OF THE CD VERSION:

The installer and installation script do not apply to purchasers of the CD.

Go to

The Screen

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

'main'

1.36 The Screen

As soon as you load OctaMED Soundstudio V1 for the first time, you'll
see that the main screen is split into four windows: Main Control
Tracker editor, Information and Tempo (Tempo not open by default).
The Notation Editor Window can be opened by holding the right mouse button
over the Display menu and highlighting the Notation Editor selection,
This new feature will be described later.

If at any time you don’t need one of the windows open, you can close it and reopen it later which helps save that cluttered effect of earlier versions. Also, the font (style of writing) used in the windows can be changed: this will be described later.

The Main Control window (Display menu -> Main Control)

This contains the playing buttons, instrument details, four window-opening buttons, Edit/Space/Chord check boxes and the octaves cycle gadget.

The instrument size box has a couple of new features. W is displayed beside the size of a 16-bit sample (W for ’word’, meaning ’16-bit’). M Cnn is shown for a MIDI instrument, where nn is its MIDI channel number; a * is also displayed if the instrument isn’t truly MIDI (perhaps it is a sample with a MIDICh value set).

SList opens the Instrument Load Window (instead of V5’s tiny button looking vaguely like a list). Edit opens either the Sample or Synthsound Editor, depending on the current instrument’s type (if the type is neither, the Sample Editor is opened).

The Tracker Editor window (Display menu -> Tracker Editor)

Home to the Tracker editor itself and track on/off buttons. The S buttons choose whether or not the track is selected (in V5 you had to cycle the upper screen’s On/Off gadget to Select). Scroll bars and a sizing gadget have been added. Tracks have more space between them, too.

The title bar shows the number of the current/last block and the current/ last command page (see §12.5). Settings menu -> Display Max Tracks has been removed as you can now select any non-proportional font (see §12.10).

Windowizing the Tracker editor has forced the bar equalizers into a separate window (select Settings menu -> Equalizers -> Bar).

The Information window (Display menu -> Information Window)

Contains the four status displays and window-opening buttons (on the upper screen’s fourth row in V5). Plus the channel mode / MIDI / Edit/Space/Chord display, the timer, and the lower screen’s free memory and general status displays.

The Tempo window (Display menu -> Tempo Window)

Here you’ll find all tempo-related gadgets. The primary tempo is called Tempo, the secondary tempo TPL (Ticks Per Line), and the BPM lines per beat setting LPB. Click the SPD cycle gadget to enable the LPB slider.

The secondary tempo, you’ll notice, is set in decimal!

There are some new gadgets too. Slow HQ is a compatibility switch for songs created with OctaMED Pro V3 and V4. In these versions, switching High Quality Mode on slowed the tempo a bit. This check box corrects this, and is saved with songs.

The Edit button opens the following and interesting window...

The Tempo Operations window

Allows conversion between the various methods of setting the tempo in OctaMED. For example, suppose that your song currently uses SPD tempo, but you suddenly decide you’d prefer BPM. In previous versions, you would:

(a) click on the SPD cycle gadget to change to BPM;
(b) adjust the Tempo and LPB sliders until, by trial and error, you find the original speed of your song.

All you need do is click on SPD -> BPM. This both changes to BPM mode and finds the values of Tempo and LPB closest to the your song’s original speed. This saves an awful lot of messing around! For example, with a tempo of SPD 32/06 (eight lines per second), clicking SPD -> BPM changes the tempo to BPM 60/08 (also eight lines per second).

Also, if you’ve ever switched from 4 channel to 5-8 channel mode, you’ll know that the song often slows down; a quick click on 4 Ch -> 8 Ch will sort this out.

So, click on one of the four top buttons to convert from:

(1) SPD to BPM   (2) BPM to SPD   (3) 4 to 8-channel   (4) 8 to 4-channel

In each case, only the Tempo slider is changed: the TPL slider is not affected. So conversion is pretty approximate, especially when converting from 4-channel to 8-channel.

If Change Commands is on, all the 0F (’set Tempo slider’) player commands in the song will be altered to reflect the new tempo. Insert Tempo Change enters the current Tempo setting as an 0F player command. (You need not have converted the tempo in order to use this button).

Set Current Tempo As Default is the one button not to do with conversion. It stores the current tempo settings in memory - SPD/BPM, Tempo, LPB and TPL. Now whenever you begin a new song, these tempo settings are recalled. To permanently use the settings as default, select Settings menu -> Save Settings.

Ok, now lets take a look at the

New Menu Items

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

The MIDI Features

Other Features

Channel Modes 5-8

Multi-Modules

Add/Del Songs

Remap Instruments

Installation

The Screen

main

1.37 New Menu Items

Project

Online help Opens the on-line help system. You can now continue using OctaMED while it's open (just send its window to the back using its depth gadget).

Command Shell Opens a Shell-like window for entering OctaMED's

ARexx Commands
, close the window by typing bye,
or by pressing Ctrl-\ or clicking its close gadget as usual.

Want more ARexx info? See the
ARexx Introduction
Audio Channels Switch this off to temporarily free the audio ↩
channels. You
can then load another music program while OctaMED is still running. Turn
Audio Channels back on when you've quit the other program (trying to play
any instrument will switch it back on automatically, if possible).

Display

Tracker Editor Now re-opens the Tracker editor if it's been closed.

Notation Editor Opens the new notation editing window and it's functions.

Main Control Re-opens the Main Control window   see
The Screen
Information Window Re-opens the Information window. (as above)

Tempo Window Opens the Tempo window.   (as above)

Block / Track / Edit

Previously, the Copy/Cut/Paste Block/Track/Range operations all used the
same copy buffer.
Now the Block, Track and Edit menus each have their own separate copy buffer.
So you can have block, track and range info all stored in memory at once.

Edit menu -> Discard Copy Buffers flushes all three copy buffers.

Block menu -> Insert/Delete Line and Split At Cursor / Join With Next
now react to line highlighting. Also, deleting a block now
removes any references to that block in the playing sequence.

Instr

Load Instrument(s) Try loading a sample, and notice that the size is now
displayed on the title bar as n [x bytes] (where n and x are numbers).
n is the sample's size, x is the actual amount of memory consumed.
(These two values will be different if a sample is 16-bit or in stereo).

Load from List Opens the Instrument Load Window. Especially useful when the
Main Control window is closed.

Save Instrument There are three new file formats: MAUD, AIFF and WAVE.
These are designed to save 16-bit and stereo samples,
(IFF 8SVX is unable to).
You can choose any one you like, but MAUD is generally used for
Toccata samples and WAVE for exporting to PC computers.
The new formats can't save loop information.

Saving a 16-bit instrument as Raw adds the header Raw16Bit (so that OctaMED identifies it as 16-bit when it’s reloaded).

Halve Loaded Samples In previous versions, a newly-loaded sample’s volume was automatically halved while in 5-8 channel mode. To prevent the halving, you had to hold down Shift while selecting an instrument loading function (e.g. Instr menu -> Load Instrument(s)).

Well, there’s a less clumsy method now! When Halve Loaded Samples is on, loaded samples will be halved (honestly!). This switch is turned on automatically when 5-8 channel mode is selected (and off when 4 channel mode selected).

So to prevent the halving in 5-8 channel mode, select this menu item (switching it off) before loading a sample.

Settings

Keyboard Shortcuts
Opens the new Keyboard Shortcuts window.

Screen Allows you to change OctaMED’s screen mode. In the sub-menu, Screen Mode opens a screen mode requester (see your Amiga’s manual for further instructions). Like WB, when on, makes OctaMED’s screen have the same resolution and number of colors as the Workbench screen.

A change of screen mode usually requires a change of font (see the next item). OctaMED’s screen is now public (name OCTAMED).

Fonts
Opens the new Font window.

Equalizers Now split into two windows. This had to be done because of the new main screen arrangement.

Miscellaneous Options
Opens the new Miscellaneous window.

Aura Sampler Options   Opens the Aura Sampler Options window.
(See also
Soundcards
)

Windows Snapshotting a window is storing its current position in memory. Normally OctaMED remembers any changes you make to the window positions automatically, but use Auto-Snapshot to switch this off if needs be.

Using the sub-menu you can snapshot the current window or all windows currently open. UnSnapshot sets the current window’s position to default when it’s next re-opened. UnSnapshot All does this for all windows in the program.

Make these window positions permanent using Save Settings. You can’t snapshot any window which uses a different set of menus from the main screen (e.g. Sample Editor, Synthetic Sound Editor).

Because of the new Window Builder (see
Fonts
), the layout of many
windows have changed quite bit. You’ll get used to them quickly.

Save Options (Project menu -> Save)

The filename text box now expands device names. For example, typing in the filename ram:song will expand it to Ram Disk:song.

File Format
Tracker module saving has unfortunately had to be removed. This is because many companies were known to be using OctaMED, but to avoid paying their registration fee their Composrs would save modules in Tracker format and make use of Protracker replay source instead. Sad (in every sense of the word) but true.

To make up for it, Standard MIDI File saving (and loading) has been added. This will allow OctaMED to be compatible with virtually all MIDI-based music programs. Only information appropriate to MIDI is saved: notes, effect player commands and SysEx messages (tempo info too).
(See
SMF Load Options
for information on loading SMF files).

Songs saved as Executable File may be loaded and ran from Workbench or a shell. The song will open a small window (displaying just its annotation text) and will begin playing: close the window to stop playing.

MMD0/1/2 now load and save text file annotation (
Song Annotation
)
and instrument output devices and MMD1 and MMD2 support command pages.

Create Icon The icons are named Module.info, Executable.info, MIDI.info and Mix.info, feel free to edit or replace them completely. You may wish to change Module.info’s default tool, for example.

Compression A new option has been added: XPK Compression. XPK is different from the other compressors in that it isn’t actually a compressor! It is instead a means of using many different compressors, all of which are in library form and are contained in the LIBS:Compressors directory. For more details, see the
XPK Settings
window.

The Buffer size nnn bytes message now also gives the percentage gain.

Settings Allows you to set PowerPacker and XPK’s options. When XPK Compression is selected, the XPK Settings window opens; otherwise, PowerPacker Settings opens. (See Powerpacker Settings)

Packed Like Calculate Size, except it works out the song size when compressed using the selected compression.

Ok, now lets take a look at the

Installation

The Screen

main

1.38 PowerPacker Settings

(click Settings in Save Options)

Set PowerPacker compression's options using this window.

Speedup Buffer The larger this buffer, the faster the compression but also the more memory-consuming. Medium is default.

Efficiency The better the efficiency, the smaller the compressed file but also the slower the compression speed. (Decompression speed isn't affected). Good is default.

(PowerPacker compression requires powerpacker.library to be in LIBS:)

XPK Settings

1.39 XPK Settings

(click Settings in Save Options)

You may select an XPK compressor and set its options in this window.

Each XPK compressor is in its own library file. For example, the SQSH (squash) compressor is in a file called SQSH.library. (I've found this compressor to be particularly good). The libraries are contained in the LIBS:Compressors directory.

When the XPK Settings window is first opened, OctaMED loads the names of all the library files in LIBS:Compressors. Select which compressor you'd like to use with the cycle gadget in the top-left corner.

The display boxes show information about the selected compressor. Use the Efficiency slider to set the compressor's efficiency (bear in mind that higher efficiency reduces compression speed).

Some XPK compressors are actually encryptors: they encode the song file using a password, which you must enter again when re-loading the song. Use the Password text box to set the password. It's ghosted if the selected compressor doesn't encrypt.

Note that by compressing using the apparently useless NONE compressor, song saving can in fact be significantly speeded up (at the expense of memory).

(XPK compression requires xpkmaster.library to be in the LIBS: directory, along with at least one XPK compressor (or encryptor) in the LIBS:Compressors directory. Only a few compressors are provided on the OctaMED disk, but if you have XPK - it's on Fish disk 754 - you're quite welcome to copy over some more. Do make sure you're using a backup of the OctaMED disk though!)

```
Sample Editor  (Display menu        -> Sample Editor
                     Main Control window -> Edit button)
```

The MIDI Features

Other Features

Channel Modes 5-8

Multi-Modules

Add/Del Songs

Remap Instruments

Installation

The Screen

New Menu Items

main

1.40 Soundcards

The Aura

This window appeared in OctaMED V5.02 under the name Microdeal Sampler Opts. It has allways been a bit naff due to HiSoft not supplying a programmers support library.....

However, give them their due, at the last minute, (two days prior to this CD being pressed), Dave Woodhouse has kindly supplied this for you all to read...

From: David Woodhouse <davew@hisoft.co.uk>
Subject: Re: OctaMED Soundstudio<>Aura Support

Hi Ray

I took another look at the Octamed program last night and got it working.
I just had to set the period value lower as the machine was a basic A1200 with no expansion.
I think it sounds a little better than before, its not brillant but it will probably do.
Perhaps it it would be a good idea to explain in the manual that any period of 360 or less with Aura on a basic machine will be unusable and anything below 200 and the machine needs a reboot.

Regards

Dave
--
David Woodhouse - davew@hisoft.co.uk

HiSoft Systems Tel: +44 1525 718181 Fax: +44 1525 713716
The Old School, Greenfield, Bedford, MK45 5DE, UK

Anyhow, like other programmers, we can do very little to fully support the Aura card as we sincerely would have liked to due to the lack of a support library.

The window allows control over the Aura 16-bit sampling / playback card. The card fits into a PCMCIA slot, so is currently only available for A600 and A1200 owners.
If you have either of these machines, though, the card is well worth buying.

A 16-bit sample must be loaded for OctaMED's Aura support to activate itself. Firstly, switch on the Active check box. (If it can't be switched on, another program is using the CIAB timers). Next, load in any sample, then open Instrument Type and click on 16-bit. You can now digitize (sample) through the card using the Sample Editor.

Other gadgets in this window:

Minimum Period sets the minimum period limit, i.e. the highest pitch OctaMED can play using the card. The smaller the period, the higher the pitch and the better the quality BUT the more processor time used.

The ideal value depends on the speed of your computer, so you have to adjust it by hand until you reach the highest possible pitch playable on your Amiga. If you set it too high, OctaMED will freeze while playing Aura samples (it will return to normal when playing has stopped).

To set the minimum period, activate Aura support as described in this section's third paragraph. Play your 16-bit sample at a high pitch (e.g. A#3), slide the minimum period down, play the sample again, and so on. You'll find that your Amiga will freeze during play if the minimum period is set too low. Set it to such a value that no freezing occurs.

You can use Settings menu -> Save Settings to save this parameter.

Single Channel Output only outputs sound using the right channel.
This saves processor time.

Fixed Output Rate forces OctaMED always to output sound at the Minimum Period rate. All notes are 'scaled' to this rate 'on the fly'.

Using this option you can play higher notes than the Minimum Period correctly (but unfortunately with degraded quality).

Any sound intended to be played through the Aura card must be of the the 16- bit type. You can easily convert an 8-bit sample to 16 bits by changing its type to 16-bit (use the Instrument Type window).

Only one 16-bit sample can be played at a time. Although they can be placed on any track, Aura samples always occupy track 3 while they're playing (this track is used for timing). So it's a good idea to enter notes played by Aura samples on track 3. (Both Aura and Toccata can't be used in 5 - 8 channel mode).

Note: The programming tool Enforcer cannot be used while outputting from Aura for technical reasons.

SMF Load Options

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

The MIDI Features

Other Features

Channel Modes 5-8

Multi-Modules

Add/Del Songs

Remap Instruments

Installation

The Screen

New Menu Items

Using Soundcards

main

1.41 SMF Load Options

(Settings menu -> SMF Load Options)

The much-requested support for the Standard MIDI File (SMF) format, used by virually all MIDI programs, is finally here! OctaMED is able to save SMF type 0 songs, and load SMF type 0 and type 1 files. This window provides many loading controls.

To load an SMF-format file, use Project menu -> Open as usual. The file may load correctly first time, but if it doesn't, try changing some of the settings in this window and reload the file (repeat if necessary). OctaMED can't guarantee a completely successful conversion, but with practice the result should be satisfactory (although conversion may take some time).

Gadgets in the SMF Load Options window are:

Max Tracks   The maximum number of tracks in the song. The default highest value of 64 may cause 'Not enough memory' reports, so reduce it if necessary. 32 works for almost all MIDI files, 16 for most.

After loading, OctaMED automatically removes unused tracks.

Lines/Block   The number of lines per block. As MIDI files are linear (in one long block), OctaMED splits the file into equally-sized blocks and creates an appropriate playing sequence, each block being played only once.

Resolution   The number of Tracker editor lines per quarter note (crotchet). So the default value of 8 assumes that there are no notes shorter than a 32nd-note (demisemiquaver).

If you find timing to be inaccurate, try e.g. doubling this value. (This will also double the number of blocks and consequently the amount of memory used).

Offset Adjust   Used to adjust minor discrepancies in timing. For example, you may find that a bass drum beat appears on lines 1, 5, 9 (etc.) instead of its correct 0, 4, 8 (etc.). In this case, set Offset Adjust to -1.

Command Pages OctaMED is able to load several MIDI commands and convert them to corresponding OctaMED player commands. You can set the command page each type of command is to use. You may place several types on the same page, but there will be the risk of overlapping commands.

If the command page number is zero (default of all except Tempo), the commands are ignored. The following command conversions are used:

Name	Command	Name	Command	Name	Cmd
Pan	0E	Note Velocity	0C	Modulation Wheel	04
Tempo	0F	Channel Volume	17	Channel Aftertch	0D
		Preset Change	1C		

Don’t Intermix Type 1 Tracks When loading SMF type 1 files, OctaMED tries to interweave the notes in each track in order to use as few tracks as possible. So it won’t normally be the case that each instrument has its own separate track, as is true for MIDI songs.

Select this check box if you’d prefer each instrument to be on its own track.

OctaMED uses instruments 01 to 0G when loading SMF-format files, one instrument for each MIDI channel. The program controls note length by setting every instrument’s Hold value to 6. Default volumes are set to 64.

Stereo support

Finally you can create and use stereo samples with OctaMED!
You’ll find the relevant gadgets at the bottom right corner.

The left-hand display box displays:

(1) 8 or 16, showing whether the current sample is 8-bit or 16-bit

(2) Mono if the sample is mono

(3) L<>R, L> or R> if the sample is stereo. L<>R means that both left and right channels are displayed in the waveform display. L> means just the left channel is displayed, R> means just the right channel.

The right-hand display box displays similar information, but it refers to the copy buffer. (Empty means the copy buffer is currently empty).

Use the S button to toggle between L<>R, L> and R> with stereo samples.

When entering notes with stereo samples, the left channel is played on tracks 0 and 3, and the right channel on 1 and 2. So to play both channels of a stereo sample at once, enter the same note on e.g. tracks 0 and 1.

Edit menu changes

Paste (Overwrite) As Paste except overwrites the existing sample contents, rather than inserting.

Erase To Start/End Erases from the sample start to Range End, or from Range Start to the sample end.

Adjust Y Opens the Adjust Y window

Centralize Centers the range about the white zero line. This allows e.g. Change Volume to be more accurate.

Three items have been moved to the new Tools menu. (Guess which ones!)

The Tools menu

Sample <-> Buffer Swaps the sample with the copy buffer.

Add Workspace Opens the Add Workspace window.

Calculate Range Time Works out the time taken to play the range at the current pitch (to the nearest .001 seconds).

Play Tune Tone Plays a sine sound at the current pitch, for instrument tuning purposes. Stop it using the space bar.

Copy Pitch Copies the current pitch to the instrument’s default pitch (in Instrument Parameters), or vice-versa.

The Loop menu

Show Loop Magnifies the loop to fill the whole display (with a bit of leeway either side).

Snapshot Loop Stores the current loop position in memory. Allows you to make adjustments to the loop, then to recall the original loop if you’re unsatisfied.

Recall Loop Retries the stored loop position.

Mark Loops the range, current display or whole waveform.

Find Zero Loop In / Out In does a 0> to the loop start pointer and a <0 to the loop end pointer. Out does the opposite.

Play Loop Plays the loop repeatedly.

Range = Loop Marks a range over the loop.

Erase Before / After Loop Removes any portion of sample existing before the loop start or after the loop end.

Pitch In Hz (Settings menu) When on, it displays the Pitch value
(near the bottom right of the main window) in Hertz - samples per second -
instead of displaying its period.

Create Icons For Samples (Settings menu) saves the PROGDIR:Icons/Sample.info icon with samples when on.

The former Pixel Density (Settings menu) window opener has been changed to Display Settings

Change Volume window changes

(Effects menu -> Change Volume)

Max Clip Allows slight clipping (only active when Check Clip is on). For example, when set to 10, the normal waveform limits are allowed to be exceeded by 10%.

Max Increases the range to the highest possible volume without introducing distortion. (It works by switching Check Clip on and setting Start and End to 999%).

Chord Creation window changes

(Effects menu -> Create Chord)

Major improvement....This window is now the tinkerer's paradise!

The bank of eight buttons in the window's center select eight different preset chords. Major, Minor and Sus 4th are three-note chords, the others are four-note.

The Inversion cycle gadget selects which note of the chord is lowest. Root is normal, First makes Chord Note 1 lowest, Second Chord Note 2 etc.

The remaining three cycle gadgets set the actual chord notes:

(1) Basic sets Chord Note 1. Major = E, Minor = D#, Sus 4th = F.
(2) Third sets Chord Note 2. Diminished = F#, Normal = G, Augmented = G#.
(3) Fourth sets Chord Note 3. Sixth = A, Dominant = A#, Major = B.

Here are some examples of chords you can create using these gadgets:

Name	Inversion	Basic	Third	Fourth
Major, 1st inv.	First	Major	Normal	- off -
Minor 7th, 1st inv.	First	Minor	Normal	Dominant
Half-diminished	Root	Minor	Diminished	Dominant
Diminished	Root	Minor	Diminished	Sixth
Major 7th, flat 5	Root	Major	Diminished	Major
Min-maj, 3rd inv.	Third	Minor	Normal	Major

Because higher notes are played more quickly than lower notes, the notes in the chord won't end together. Switching Erase Trailing Notes on removes the part that doesn't contain all notes in the chord, shortening the sample but ensuring that the notes do end together.

Full Volume doesn’t divide each value by the number of notes in the chord, increasing the volume. Distortion may occur.

Play Chord previews the chord by playing one chord note through each sound channel (the final chord will be played through only one channel). Click Mute or press the space bar to stop the chord.

A tip to improve quality: before creating the chord, change the sample to 16-bit (click 16-bit in Instrument Type).
Then click Create Chord, click Max in Change Volume, then change the sample back to 8-bit (click Sample in Instrument Type).
When you’ve completed reading the ARexx instructions,
try turning it into an
ARexx Script
Pixel Density window changes

(Effects menu -> Display Settings)

(Now called Display Settings)

Minimum Zoom is the minimum value of Display (top left corner of the main sample editor window) in bytes.

Dragging the zoom slider (far right of sample editor) upwards usually zooms in towards the middle of the sample. With Center Zoom Slide on Range on, however, it zooms towards the middle of the current range. Try switching this on, marking a range, and using the zoom slider. Quite neat really.

When Fast Graphics is on (default), the waveform display updates much faster than usual. However, the function occupies about 5K of chip memory, so switch it off if memory is very tight.

The Adjust Y window

(Edit menu -> Adjust Y)

Use this window to adjust the vertical position of the range. Type the required amount of adjustment into the Adjust by numeric box. The number can be -128 to 127; a negative number adjusts downwards.

Clicking Calculate Average Deviation works out the required adjustment to centralize the range; this is probably the most frequent use for this window. To centralize the whole sample, use Edit menu -> Centralize in the sample editor.

Click Adjust to adjust the range. (As usual, shift-click it to close the window as well).

The Add Workspace window

(Tools menu -> Add Workspace)

With this window, you can add extra space (of zero volume) to the sample’s start or end. Enter the number of bytes of space to add into the Bytes box.

Alternatively, use the two sliders to calculate the number of bytes:

(1) Factor calculates the number of bytes as whole multiples of the sample size. For example, if the sample size were 2000, a Factor value of 3 would add 6000 bytes to the sample.

(2) Use Adjust to add that little bit extra on to the Factor value. For example, for a Factor value of 2½, set Factor to 2 and Adjust to 50%.

A quick way to add as much space as the sample itself, doubling the sample size, is to set Factor to 1.

Usually you'll probably want to add space to the end of the sample, but use the Add to gadget to add space to the start.

Click Add Workspace to add the space.

Other sample editor changes

An important change: Operations will now act on the whole sample if the range is one byte long (i.e. Range Start = Range End). So an alternative to clicking Range All is to click once anywhere on the waveform, or to leave Range Start and End set to zero.

The scroll bar on the right of the waveform allows magnification of the display. Drag it upwards to zoom in, downwards to zoom out.

Shift-clicking the arrows beside Loop Point (bottom of window) sets the relevant loop pointer to the start or end of the sample. (By the way, dragging the loop pointers is now infinitely faster!)

Input Map Editor

(Display menu -> Input Map Editor)

The Functions window has been removed! There's now a much more flexible way of setting a key's function.
It works like the new Keyboard Shortcuts window (see
Keyboard Shortcuts
),
in that you specify the OctaMED command required to perform the function.
You can also run any ARexx script using the keyboard.

The cycle gadget at the top of the Action area states what general action a key should have. Select None if the key is to enter a note as normal.

I'll briefly list the commands required to perform the functions given in the old V5's Functions window.
Make sure the cycle gadget shows OctaMED Command, then type the required command into the Command text box.

Functions	Command	Functions	Command
Play Song	pl_playsong	Prev Block	ed_gotoblock prev

```
Play Block | pl_playblock            || Next Block | ed_gotoblock next
Cont Song  | pl_contsong             || Edit       | op_set edit toggle
Cont Block | pl_contblock            || Space      | op_set space toggle
Stop       | pl_stop                 || Chord      | op_set chord toggle
Curs Up    | ed_advanceline up spc   || Del        | ed_clearnotecmd note
Curs Down  | ed_advanceline down spc || Hold (-|-) | ed_enternote hold
Curs Left  | ed_movecursor left      || Prev Inst  | in_select prev
Curs Right | ed_movecursor right     || Next Inst  | in_select next
Prev Track | ed_gototrack prevnote   || First Line | ed_advanceline fkey 6
Next Track | ed_gototrack nextnote   || Last Line  | ed_advanceline fkey 10
```

Channel Modes 5-8

Multi-Modules

Add/Del Songs

Remap Instruments

Installation

The Screen

New Menu Items

Using Soundcards

SMF Load Options

1.42 Song Annotation

(Song menu -> Set Annotation)

Greatly improved: you can now attach a whole text file to a song!
So no need for those 'readme' files you always find on module disks.

Create the file in an external text editor, then click Load File to load it.
Save Text saves the text under a chosen name, and Discard Text removes the
text from memory. The text is (of course) saved with songs.
By clicking on the "Show Upon Loading" selector, you will ensure that
your saved text is opened immediately the module is loaded back in.

Note that tab characters (ASCII code 9) can't be used in the text.

Block Properties

```
(Block menu         -> Set Properties
                       Information window -> shift-click B button)
```

Cmd Pages sets the number of player command pages in the current block.
Using this feature, notes can have more than one player command attached to
them. For example, if you wanted the note G-2 to have a sample offset
of $500 (hex) and to play at volume 32, you would use this:

```
   Command page 1:  G-2 11905          Command page 2:  G-2 10C32
```

Press Shift-Tab to cycle through command pages in the Tracker editor. So to
enter the above, first increase Cmd Pages to 2 (using the > button). Now
enter G-2 11905 in the Tracker editor, press Shift-Tab, then enter 0C32.

Note: two pitch-changing commands compensate each other
(e.g. 0205 + 0102 = 0203).
Several 00/04/14 commands can be used together, but only the one on
the last page counts. 03 must be on page 1 if used with held (-|-) instrs.

The Tracker editor's title bar shows the current page and the total number of pages in the block. Editing operations such as Cut / Copy / Paste Range act on all command pages at once. The number of pages a block can have is virtually unlimited (well, all right, the maximum is 32767!).

Instrument Type

(Instr menu -> Type
Main Control window -> Type button)

Normal Amiga samples and synthetic sounds are 8-bit, but if you have one of the supported 16-bit cards (see
Soundcards
) allow you to work with
much higher quality samples.

This window provides some 16-bit options for use with those cards.

Click on the 16-bit radio button (bottom left) to make the
current sample 16-bit.
A 16-bit sample can be converted to 8-bit by clicking on Sample
(but you'll lose sound quality).

The Stereo check box is checked when the current sample is in stereo.
Click on it to change a mono sample to stereo and vice-versa (you'll need some free memory for mono -> stereo, in fact double the sample's size: press Ctrl-F and check the largest Chip number).

Output Device selects whether the current sample should be played through the Amiga or through one of the chosen
Soundcards
(The Toccata option is only shown if a Toccata card is actualy connected).

Transpose

(Edit menu -> Transpose)

Just the one check box added, Instrument Slots. When on, the three Change Instrument buttons - Change, Swap, Delete - act on the actual instruments themselves rather than notes played by the instruments.

Try it yourself! Load an instrument into slot 01 and another into 04. Now select 01 for Source (i.e. select instrument 01 and click Source in the Transpose window), and choose 04 for Destination. Finally switch Instrument Slots on and click Swap. Notice that instrument 01 has swapped slots with instrument 04.

Delete flushes the source instrument from memory.

Keyboard Options

Creating a Song

SMF Load Options

main

1.43 Keyboard Options

(Settings menu -> Keyboard Options)

All the gadgets involving keyboard shortcuts have been moved to the Keyboard Shortcuts window. These include the two numeric keypad cycle gadgets, and the Space = Del and Protracker Shortcuts check boxes.

The new F6-F10 = Highlights check box affects the F6 - F10 keys (used to move to certain lines in the Tracker editor). It sets the positions moved to by these keys to the first five highlighted lines in the current block. Useful for widely-spaced blocks with few samples in them (the beginning of a rave track, for example?).

Poly Play: When on (and edit mode is off), playing notes using the keyboard will play the notes through alternate sound channels. This gives a kind of resonating effect. Try it!

Keyboard Shortcuts

Keyboard Directory

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

1.44 Keyboard Shortcuts

(Settings menu -> Keyboard Shortcuts)

This window allows you to change and add to OctaMED's keyboard shortcuts.
You can set practically any key combination to operate almost any OctaMED
feature. You can also run any
ARexx Script
script
using the keyboard.

The basic steps to setting a keyboard shortcut are:

Step 1) Create space for a new shortcut and name it.

Step 2) Set the shortcut's key combination.
Step 3) Enter the operation to be performed by the shortcut

Step 1

On the window's left is a list of all shortcuts. Use this list to accomplish step 1) above. Use Ins. New or App. New to insert or append a blank shortcut. Now click inside the Name text box and rename it (maximum name length is 99 characters).

(You may also delete a shortcut in this section. Select the shortcut by clicking on its name in the list, then click Delete)

Step 2

Here you decide which key combination triggers the shortcut. You can set:

(a) A qualifier - the key(s) you hold down (e.g. Shift, Alt, Amiga, Ctrl)
(b) The actual key you press together with the qualifier (e.g. the V key)
(c) Whether the caps lock should be on or off (or to ignore the caps lock)

Set the qualifier using the first four cycle gadgets. The Shift, Alt and Amiga gadgets each have the following options:

Ignore Ignores the status of the Shift/Alt/Amiga key. That is, it doesn't matter whether Shift/Alt/Amiga is held or not.

Either Either of the Shift/Alt/Amiga keys (left or right) must be held.

Left The left key must be held. (The right one can optionally be held)

Right The right key must be held. (The left one can optionally be held)

Both Both left and right keys must be held (rare!).

None Neither key must be held (not quite the same as Ignore).

Left Only Only the left key must be held (not the right).

Right Only Only the right key must be held (not the left).

The Control and Caps Lock cycle gadgets have Ignore, On and Off options. Ignore ignores the status of the Ctrl/Caps Lock key. On means it must be on (or held in Ctrl's case), Off means it must be off.

Set the key you press together with the qualifier using the Mapped and Raw gadgets. If the key is mapped (i.e. a letter, number or symbol appears on the screen when you press it), click inside the Mapped gadget and type it in.

If the key is not mapped, you'll have to type its raw-key code into the Raw gadget. Here's a list of raw-key codes you may need: (NK = num. keypad)

Key	Code	Key	Code	Key	Code	Key	Code

Esc	69	F8	87	NK5	46	<down>	77
F1	80	F9	88	NK6	47	<left>	79
F2	81	F10	89	NK7	61	<right>	78
F3	82	NK0	15	NK8	62	Backspace	65
F4	83	NK1	29	NK9	63	Tab	66
F5	84	NK2	30	NK.	60	Return	68
F6	85	NK3	31	Enter	67	Del	70
F7	86	NK4	45	<up>	76	Help	95

Step 3

Here you decide what effect the key combination will have. Use the cycle gadget to choose from four possible effects:

(a) OctaMED Command — The option you'll use most often: it executes an OctaMED ARexx command. Type it into the Command box, together with any required parameters. To execute more than one command, use the OP_MULTICMD command.

(b) Execute ARexx File — Runs an external ARexx file. Type its name into the Command box (unless the file is in PROGDIR: or REXX:, specify the full path name). The filename should end in .omed.

(c) Ext. ARexx Command — Sends an ARexx command to another program. Type the command into Command, and the program's ARexx port name into ARexx Port.

(d) Launch Program — Loads and runs a program from disk. Type its name into Command.

The maximum length of Command is 255 characters.
Note: To use Execute ARexx File or Ext. ARexx Command successfully, make sure you've run RexxMast.

Other Gadgets and Notes

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

1.45 Other Gadgets and Notes

The Window text box allows you to create shortcuts that only work ↩ in a
particular window. In this way, the same key combination can be used for different purposes in different windows.

Select required shortcut, then type the window’s ARexx name into the Window.
If the same key combination is defined more than once as ’global’
(not window-specific) and window-specific, the latter has priority.

Load and Save selectors load and save keyboard shortcut files.
The default name is PROGDIR:Soundstudio.defkeyboard.
(PROGDIR: is the directory in which the OctaMED program resides).
When loading, OctaMED will ask whether you’d like to keep the existing
shortcuts (adding the new file to the end of the list), or to replace them.
(removing them from memory)

Once set, you may alter a shortcut’s name, key combination and effect.
Simply click on its name in the list and make the desired changes.

Keyboard Directory

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

The MIDI Features

Other Features

Channel Modes 5-8

Multi-Modules

Add/Del Songs

Remap Instruments

Installation

The Screen

New Menu Items

Using Soundcards

SMF Load Options

Keyboard Options

Other Gads/Notes

main

1.46 Keyboard Directory

```
                You’ll find many example shortcut files in the Keyboard directory
(or in the Keyboard directory of your hard disk or non-bootable disk).
Simply click on Load and select the appropriate file.

Included in the directory are definitions for the shortcuts removed from
Keyboard Options.
The four Pad files contain the four numeric keypad settings.
PTKeys defines Protracker shortcuts.
(See the next section for implementing Space = Del)

(1) LAmiga contains many Left Amiga shortcuts which open various windows
    (to accompany the Right Amiga shortcuts).

(2) Return effectively adds a new feature: press Return to close the
    currently active window (this doesn’t apply to the three main screen
    windows).

(3) In V5, by switching the caps lock on you could enter programmable keys
    without holding Shift and use the playing sequence without holding Ctrl.
```

Load the Caps file to resurrect this feature.

If you'd like any of these shortcuts to be default, make sure you click Yes in the 'Retain existing shortcuts?' requester when loading the appropriate shortcut file. Then save the shortcuts using the Save button.

A fair amount can be learnt about keyboard shortcut creation by studying these example files. Or try adding the following shortcuts yourself...

Some more example keyboard shortcuts

To add one of these shortcuts, click on Ins. New and type its name into the Name box. Now type the key into the Mapped box and press Return, (or type the key's raw-key code into Raw if necessary), now set the correct qualifier, and finally type the shortcut command into the Command box.

(1) Name: Space = Del.   Key: space bar (no qualifier).
Command: pl_playstatecommand "pl_stop" "ed_enternote 0"

You'll have to delete the Space (STOP) shortcut before you can use this. Find its entry in the list, click on it then click Delete.

(2) Name: Del with Caps. Key: Del (Caps Lock = On).
Command: ed_clearnotecmd note cmd

With the caps lock on, the Del key will now act like Shift-Del. [Requested by Chris Wright]

(3) Name: Ctrl-9 (90% Vol). Key: 9 (Control = On).
Command: sa_changevol 90 90. Window: sampleeditor (type into Window box)

With the Sample Editor active, press Ctrl-9 to reduce the range's volume by 10%. Try creating a similar shortcut (perhaps Ctrl-1) to increase the range by 10%: change the 90 90 to 110 110. [Requested by Dave Sullivan]

Note that this shortcut won't work in the Change Volume window. You'd have to create an identical shortcut with Window set to sa_changevolume.

(4) Name: Ctrl-P (PolyPlay). Key: p (Control = On)
Command: op_set polyplay toggle

Toggles Poly Play (Keyboard Options window) on/off.

(5) Name: Sh-Ctrl-C (App blk + copy). Key: c (Control = On, Shift = Either)

```
Command: op_multicmd "op_update off" "rn_copy block"
                     "ed_newblock clonecurr last" "ed_gotoblock last"
                     "rn_paste block" "op_update on"
```

Creates a new block, and copies the current block into this new block.

(6) Name: Ctrl-R (Read file). Key: r (Control = On)
Command: c:ppmore ps OCTAMED. Cycle OctaMED Command to Launch Program

Reads a text file by running PPMore (if it exists) on OctaMED's screen.

Using

New Menu Items

Using Soundcards

SMF Load Options

Keyboard Options

Keyboard Directory

Other Gads/Notes

main

1.47 Fonts

(Settings menu -> Font)

Using this window you may change the fonts (styles of writing) used in three different areas of OctaMED.

This feature has been made possible by a new way of drawing windows, also written by OctaMED's author, called Window Builder. Windows can now use fonts of any size (within reason!) because OctaMED adjusts the size of gadgets and windows to suit the new font size.
If the window wouldn't fit on the screen using the selected font, however, it's drawn in the normal Topaz 8 style.

In fact, OctaMED calculates a window's gadget sizes and positions whenever it's opened, so you may notice a slight delay when you open a window. Also, as a little spin-off the windows' layout has been changed a bit and they all look a bit nicer and centralize on the screen etc. (Lovely!)

You change the font used on the screen (i.e. title bars, menus etc.), in windows and in the Tracker editor separately. Click on the appropriate GetFile gadget (left side of window) to select fonts, or switch the relevant Default check box on to select the system default font (Topaz 8 unless you've changed it from the Workbench).

On clicking a GetFile gadget, OctaMED loads the names of all the fonts currently in your FONTS: directory. If you'd like to use a font from another disk, the easiest way is to copy the font over to your SYS:Fonts dir
Here's how to do it:

(a) On the Workbench screen, open a shell (Project menu -> Execute Command, then type newshell and press Return)

(b) In the shell that appears, type makedir fonts:<font>. For example, if you wish to use Diamond, type makedir fonts:Diamond.

(c) Type copy <fonts_disk>:Fonts/<font>/<point_size> fonts:<font>. For example, if your fonts disk is called Art, and the font you wish to use is Diamond 20, type copy Art:Fonts/Diamond/20 fonts:Diamond.

(d) Finally, type copy <fonts_disk>:Fonts/<font>.font fonts:Diamond. So you might type copy Art:Fonts/Diamond.font fonts:Diamond.

You may use any font except ColorFonts and AGFA Compugraphic^1 scalable fonts (used with many wordprocessors). Also, only non-proportional fonts may be used in the Tracker editor, and proportional fonts may cause slight display problems under Kickstart 2.04.

If the font is too large, you may wish to change the screen mode to a higher resolution e.g. 640 x 512 (select Settings menu -> Screen -> Screen Mode). If you like a large font in the Tracker editor in 4-track blocks, try loading the widetop font found in this disks fonts drawer into Editor Font. (Widetop is installed on a bootable floppy or optionally on a hard disk.).

You can, of course, use the special ^{1}Compugraphic font supplied with this program, to print your scores via the notation editor window in this much nicer format.

@{ "Miscellaneous Options " Link Miscellaneous_Options}

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

1.48 Miscellaneous Options

```
                (Settings menu -> Miscellaneous)
```

All sorts of little settings here. Both H -> B and Workbench Open/Close have moved to check boxes. In the latter case, this means that by switching the check box off and saving settings, OctaMED will attempt to close Workbench every time it loads.

All other options are new. Here they are:

Overwrite Requesters   Enables you to switch off those sometimes irritating 'File already exists. Overwrite?' requesters.

Warn if Disk Full

When on, a requester appears if the song you're trying to save probably won't fit on the disk. Be warned, though: it's impossible to predict this accurately, and the file might not fit even if the requester doesn't appear.
Also, the requester always appears when saving to RAM:, so just ignore it in this case.

(Note that this only applies to saving songs)

Size-Only Window Zoom

Normally, clicking a window's zoom gadget shows only the window title bar and moves the title bar to its default position. (For a demonstration, drag any window to a different position, then click on its zoom gadget. It moves back to its original place, yeah?)

With this check box on, the window won't move back to its default position, it will stay right where it is.
But only under Kickstart 3.0 or higher...

Use ReqTools

Uses the ReqTools normal / file / screen mode / font requesters instead of ASL. Requires reqtools.library. If you wish to reload the available fonts (see Fonts), you should switch this off.

(Note that if asl.library isn't available on startup but reqtools.library is, OctaMED will automatically use ReqTools, regardless of the state of Use Reqtools).

Load Instr From MainCtrl

When on, typing a name into Main Control's instrument name text box attempts to load the file.
When off, the instrument is simply renamed.
Even when on, press Alt-Return in the text box to rename the instrument.
Such fine detail in this program...

Default Volume Mode

Selects whether the cycle gadget in Song Options displays Decimal or Hex Volumes, in other words, cycle this gadget to Hex and save settings if you prefer using hex volumes with command 0C.

Gadget Shortcut Qual

Choose the qualifier used with gadget shortcuts: Left Alt (default), Right Alt, Left Amiga or none at all (i.e. gadget shortcuts are disabled). When using Left Amiga, beware of the system shortcuts L.Amiga-N and M!

Actually, gadget shortcuts now apply to:

(a) Text and numeric boxes (they activate the box)
(*b) Cycle gadgets (hold Shift with the qualifier to cycle backwards)
(*c) Sliders (hold Shift with the qualifier to decrease the value)

as well as action buttons and check boxes.
Those marked with a * are only available under Kickstart 3.0 or higher.

Default HQ Mode   Sets the default High Quality Mode (Song Options).

Default Slow HQ Mode   Sets the default Slow HQ (Tempo, see §10.4).

Help Viewer

Selects whether to use AmigaGuide, Hyper or XPKGuide to display the on-line help system.
The advantage of the latter two programs is their ability to load compressed files.
With Hyper, you may compress OctaMED.guide with PowerPacker; with XPKGuide, with an XPK compressor.

Make sure the directory containing Hyper or XPKGuide is in the command path (use the Path command). It's probably safest to keep them in C:.

Changes to Other Windows

Creating a Song

Editing The Song

Linking blocks

Adding new blocks

Block contents

Using The Blocks

Saving the Song

Player Commands

Slide creation

Transpose Window

Track Selection

Programmable Keys

Numeric Keypad

The Sample Editor

Magnification

Special Effects

1.49 Changes to Other Windows

Save Timer: Active switches the save timer on/off. If Open Save Window is off, the Save Options window only opens the first time the time elapses. After this, the song is saved using the current save settings.

Synthsound Editor: The program-related gadgets are now in a new SynthEd Program window. You can close it if not required, and reopen it using the Program button (slightly below center of main synth editor window).

There’s a new Project menu item, Reset Temp Wave. It ’resets’ the right-hand (or temporary) waveform, clearing it and setting its length to 128.

Sample List Editor: After clicking Add Dir and choosing a directory, there’s a new Top Of The List button in the requester. Revolutionary! Also, the append/replace requester appearing in V4 when loading a new list has been resurrected. In the menu, Clear List clears the entire sample list (after confirmation).

Song Selector: The display box has changed into a text box.

Playing Sequence:

Two new buttons, Append and Append Curr, work like Insert and Insert Curr except they add the new entry to the bottom of the sequence.
This should make creating playing sequences from scratch a bit easier.

Song Options:

If you switch to e.g. 7 channel mode, and some of the blocks in your song have less than 7 tracks, OctaMED will now offer to add the missing tracks to these blocks.
The High Quality Mode gadget can now be switched on in 4-channel mode, but it has no effect on sound quality.

Relative Track Volumes:

Just one layout change, only 8 tracks are now shown simultaneously. (16 previously)
If you use MIDI you may need to use the two arrow buttons a bit more frequently to access tracks higher than 7, but we think the window looks a lot neater for the change.

Block List:

There’s a new Show Unused check box. When on, any block that hasn’t yet been used in the playing sequence is marked with a *.

Instrument Parameters:

When the new Disable check box is on, OctaMED pretends the current instrument isn’t there! All notes played by it in the Tracker editor are completely ignored.
This was added at the request of a user who asked for it for what he calls ’multi-tracking’.
When recording songs on to tape, he likes to record selected instruments on to each track of his tape recorder.

The instrument selection gadgets now update much more quickly when, for example, using the selection slider. Also, the Vol slider now displays its value in decimal and hex (as in the very old V4).

Instrument Load Window: The current instrument number is now shown beside Prev Inst.

Mouse Options: Position Cursor now reacts to clicking on command digits.

Palette Window:

The new check box WB Palette (on by default) uses the palette used by Workbench. Switch the check box off to adjust the color sliders.

Miscellaneous improvements (not concerning windows)

(a) The right mouse button no longer shrinks / expands the currently active window as in V5. Instead, it displays a set of menus as usual. It can also once again be used to mark a range in the Tracker editor (use the Mouse Options window to select this), and menu shortcuts do work over the editor now.

(b) OctaMED now takes advantage of Kickstart 2.04’s improved loading and saving routines, so disk operations should be faster. (We have not used later Kickstarts to enable those with older machines to use this new version, providing they have enough memory and a nice fast accelerator)

(c) The bug which caused OctaMED to sometimes crash on quitting has finally been fixed. Tieijo had in fact made a typing error!   :)

Plus, of course, far too many other bugs fixed and minor enhancements made to even think about listing them all!

1.50 Using the monitor for digitizing

Monitoring/Digitizing

Any user who tried sampling using V5 will be relieved to hear it’s been very dramatically improved!

Instead of the quirky, slow window, OctaMED now opens a black area at the bottom of the screen. Almost all multi-tasking is disabled (high priority operations such as mouse pointer moving and disk access are still available).

Controling the monitoring / digitizing process is now much how it was before V5. When monitoring, click inside the black area to quit. You may notice some crackling, this is quite normal and won’t affect digitizing (thankfully!). When digitizing, click to start the process or quit using the right mouse button.

The small gadget at the top left corner of the black area selects whether input should be accepted from the left or right channel (half) of a stereo sampler. (If you own a mono sampler, ignore this gadget).

You can now sample from an Aura, Toccata, MaestroPro & Delfina 16-bit card.

Simply type the required sample length into Buffsize. Then in the Instrument Type window, set the current instrument’s type to 16-bit and set its output device to Aura or Toccata etc, providing you have the card fitted of course, now click Monitor or Digitize. Because of the new 16-bit support, the Mix routines and Sample to Fastmem additions there is now no limit on sample length, the only limit being of course, the amount of memory that you have.

```
The Sample Editor

Magnification

Special Effects

Sample Range Ops

Sample Loops

The Sample List

Inst Features

The MIDI Features

Other Features

Channel Modes 5-8

Multi-Modules

Add/Del Songs

Remap Instruments

Installation

The Screen

New Menu Items

Using Soundcards

SMF Load Options

Keyboard Options

Keyboard Directory

Other Gads/Notes

Using Fonts

Miscellaneous Opts

Window Changes

Soundstudio Features

main
```

1.51 tutorial

Soundstudio Fähigkeiten

Dieses Material wurd von Teijo - Ed - Ray geschrieben und ist Copyright 1996

Diese Datei listet einige der Änderungen und neuen Fähigkeiten, die in OctaMED Soundstudio V1 hinzugekommen sind (es kommt noch mehr hinzu).

(Wenn Sie V6 nicht kennen, sondern nur V5, werden Sie Überraschungen erleben!)

Andere Eigenschaften

1.52 Misch-Routine

Dies ist wohl der auffäligste Unterschied: Statt der alten 4-Kanal Mischroutine, das stark an die Audio-Hardware des Amigas angebunden war, ist die neue Mischroutine jetzt Hardware-unabhängig und es können verschiedene Ausgabe-Optionen ausgewählt werden. Unterstützte Ausgabe-Geräte sind der Amiga (8- und 14-Bit), Toccata (16-Bit), MaestroPro (16-Bit) und Delfina (16-Bit). Es ist auch möglich die digitale Ausgabe direkt auf Disk zu speichern, wo bei die Bitzahl als auch die Sampling-Frequenz variiert werden können.

Die Misch-Routine erlaubt bis zu 64 unabhängige Audio-Kanäle, unterstützt sowohl 8- als auch 16-Bit samples, benutzer-definierte Misch-Frequenzen (abhängig von Ausgabe-Gerät), Spurweises Panning, das rückwärts abspielen von Samples, Ping-Pong Loops, das Abspielen von Samples aus dem Fast RAM, einen 6-Oktave umfassenden Bereich, OctaMED Synthsounds, Stereo und Mono Modi, Echtzeit-Echo/Kreuz-Echo, Stereo-Seperation und hoch qualitative Interpolation um die digitalen Daten auf Disk zu speichern.

Die Mischroutinen wurden in optimizierten Assembler-Code geschrieben, sowohl für 68000 als auch 68020+, auch wenn die Mischroutine auf einem 68000 recht langsam ist.

Ein Beispiel, der A3000/25 kann ca. 10 Kanäle bei 48kHz durch die Toccata abspielen.

Um die neue Misch-Routine noch attraktiver zu machen, kann OctaMED S3M und FastTracker 1.0 Module laden.

Wie schon erwähnt können die Module (oder Teile davon) direkt auf Disk gespeichert werden. Dies kann dazu benutzt werden, um Schlagzeug-Loops zu erzeugen, die sonst mehrere Spuren benötigen würden. Der erzeugte Sample benötigt dann nur noch einen Kanal. Eine spezielle "Smooth" Funktion ist verfügbar, wenn man auf Disk abspeichert. Es filtert unerwünschte Frequenzen weg, wodurch die Qualität zunimmt. Auf einem 68060 kann man es auch zum Echtzeit-Abspielen benutzen :-)

Die Tatsache, dass die Audio-Daten durch den Prozessor gemischt werden, eröffnet erstaunliche Möglichkeiten. OctaMED SoundStudio bietet digitales Echo und Kreuz-Echo in Echtzeit an. Ich hoffe, dass ich den Bereich der DSP Funktionen in der zukünftigen PC Version ausbauen kann.

Das Abspielen von 8 oder mehr Kanälen....

Wählen Sie "Optionen Einstellen" aus dem Lied-Menü und klicken Sie auf 1-64 Ka. Wählen Sie dann "Misch-Routine" aus dem Einstellungs-Menü und setzen sie die Max. Anzahl von Kanälen auf die gewünschte Zahl. Das Laden eines S3M-Moduls oder ein OctaMED Moduls (es gibt nur sehr wenige im Augenblick) setzt diese Einstellungen automatisch.

Das Laden von Tracker oder MED Modulen in Misch-Modus...

Öffnen Sie das Einstellungen-Menü und wählen sie "Verschiedenes".
Klicken Sie dann auf "Benutze Mischen (MMD)" oder "Benutze Mischen (Tracker Mods) ↩
".
Dann wird Ihr altes Mod direkt in die neuen Misch-Routine geladen.

Misch-Routine

Noten-Editor

main

1.53 Noten-Editor

Der Noten-Editor, den es in V6 nicht gab, ist zurück. Er wurde stark verbessert. Hier einige Highlights:

- Bis zu 16 Partituren, deren vertikale Position und Breite angepasst werden können.

- Vorzeichen können frei ausgewählt werden (statt wie in V5 nur 2 Möglichkeiten).

- Ein Compugraphic Noten-Font befindet sich im Lieferumfang, um die Druckqualität zu erhöhen.

Einige Leute wollten einen "Professionellen" Noten-Editor, aber ein solcher Editor wird es wohl in der Amiga OctaMED Version nicht geben. Es ist quasi unmöglich einen professionellen Noten-Editor mit einem Tracker-Programm zu kreuzen. Das Ergebnis wäre etwas, womit niemand zufrieden wäre.

Wer einen professionellen Noten-Editor braucht, kann die OctaMED-Dateien als MIDI exportieren.

Der Noten Editor wird durch das Anzeige-Menü geöffnet. Er bietet eine alternative Anzeige des Liedes an. Es benutzt die Standard-Notation, statt der computerfreundlicheren Notation aus dem Tracker-Editor. Die Musik kann ausgedruckt werden und auf einem anderen Musik-Instrument wiedergegeben werden.

Der Noten-Editor ist stark an den Tracker-Editor gebunden. Eigentlich zeigen beide das gleiche an, aber auf verschieden Weise. Wenn Sie eine Note in den Noten-Editor einfügen, wird diese Note auch in den Tracker-Editor eingefügt. Die OctaMED-Befehle können allerdings nur im Tracker-Editor eingegeben werden.

OctaMED ist vor allem ein Sequenzer, der auf dem Tracker basiert und der Noten-Editor ist nur als Zugabe gedacht. Dennoch ist er viel besser als der aus der Version 5.

Grundlegende Operationen

Voreingestellt ist, dass man 2 Notenreihen sieht. Sie können das über das Menü ändern.

Bevor die Noten angezeigt werden, müssen sie diese den Tracker-Spuren zuordnen.

In der Titelzeile des Noteneditors, werden sie etwas sehen, wie Block 0/2 - Zeile 0 - 15. Dies bedeutet, dass der Editor gerade die Zeilen 000 bis 015 des Blocks 0 anzeigt, der letzte benutzte Block ist 2. Eine Tracker-Editor Zeile wird als einen 16tel Note angezeigt. Eine Viertel-Note ist also vier Tracker-Zeilen lang.

Um Noten einzugeben, versichern Sie sich, dass Edit eingeschaltet ist (Hauptkontrolle) und wählen Sie eine Note aus dem Werkzeug-Fenster aus. Klicken Sie nun an die Stelle, wo die Note erscheinen soll.

Wenn sie die Maustaste gedrückt halten und über die Partitur ziehen, können sie die Noten entsprechend der Position hören. Die Note wird übrigens im Werkzeug-Fenster angezeigt, darüberhinaus findet man dort die Tracker-Editor Zeile.

Das Projekt Menü enthält folgende Einträge:

Drucken Öffnet das Druck-Fenster um bestimmte Teile auszudrucken. Hier kann man verschiedene Qualitäts-Stufen auswählen.

Noten-Editor verlassen Schliesst alle Fenster, die mit dem Noten-Editor ↩
zusammenhängen.

Das Fenster-Menü hat die folgenden Fenster.

Spuren zuweisen Hier können sie einstellen, welche Spur welcher Partitur ↩
entspricht.

Einstellung der Staff Sie können Partituren hinzufügen und entfernen

Vorzeichen Setzen Sie die Vorzeichen hier.

Werkzeug-Fenster Normalerweise offen. Wählen Sie hier die Noten aus.

Einstellungen anzeigen Ändert die Breite, ob die Namen angezeigt werden, die ↩
Anzahl der

Vorzeichen etc.

Das Druck-Fenster

Hier können sie ein Lied in der Standard-Notation ausdrucken.

Zuerst wählen sie den Anfangn und das Ende des Ausdrucks aus. Mit den Knöpfen Aktuell kann man den aktuellen Block als Anfang oder/und Ende angeben.

Wählen sie dann die Ausdruck-Auflösung aus. Die erste Option benutzt den standard Amiga-Font, die anderen drei benutzen den Compugraphic Font. Je höher die Zahl, desto besser die Qualität, aber desto langsamer der Ausdruck.

Jetzt drücken Sie Drucken. Nach kurzer Zeit erscheint ein Requester. Klicken Sie Stop, um das Drucken abzubrechen.

Das Fenster Spuren Zuweisen

In diesem Fenster kann man die OctaMED Spuren den Partituren zuweisen.

Jede Zeile des Fenster weist jeder Partitur eine bestimmte Spur zu.

Voreingestellt ist 0, das heisst, das keine Spuren angezeigt werden. Wenn sie Spur 2 auf der ersten Partitur anzeigen wollen, dann setzen sie den Spur 2 Regler auf 1. Falls Spur 2 leer sein sollte (im Tracker Editor), dann wird eine Pause erscheinen, andernfalls die Noten aus Spur 2.

Sie können soviele Spuren wie sie möchten in einer Partitur anzeigen, aber mehr als 2 machen mehr Chaos als Sinn.

Die Hoch und Runter Symbole schieben die Spurnummer hoch und runter. Benutzen Sie diese Symbole, wenn sie mehr als 8 Spuren in ihrem Lied haben.

Das Vorzeichen Fenster setzt die Vorzeichen des Liedes.

Benutzen sie die beiden Schieberegler um den Takt zu ändern, voreingestellt ist 4, das heisst 4/4 Takt. Der obere Schieberegler kann einen Wert zwischen 1 - 8 haben, der untere 1, 2, 4 oder 8. Komische Takte, wie 5/8 und 7/1, wie auch standard wie 3/4 und 2/2 sind möglich.

Um die Vorzeichen zu setzen, benutzen sie die unteren Schieberegler. Er zeigt die Anzahl von b’s und # an.

Das Werkzeug-Fenster

Benutzen sie dieses Fenster um die Noten oder die Pausen auszuwählen. Es wird automatisch zusammen mit dem Noten-Editor geöffnet, jedoch jederzeit geschlossen und wieder geöffnet werden.

Wenn die Note nicht sichtbar ist, dann kann man sie auch direkt eintippen, indem man ihre Länge angibt. Eine Tracker-Editor Zeile ist äquivalent zu einer 16tel Note. Noten, die kürzer als ein 16tel sind, sind daher nicht erlaubt.

Bewegen Sie nun den Mauszeiger über einen der Partituren. Die ausgewählte Note oder Pause erscheint. Zusätzlich wird die entsprechende Tracker-Zeile angezeigt. Drücken Sie nun die Maustaste. Jetzt wird zusätzlich noch die Note angezeigt, die der Position entspricht. Lassen Sie die Maustaste wieder los, um die Note hinzuzufügen.

Die Symbole ganz rechts werden benutzt, um die Partitur und die Tracker-Spur auszuwählen, die man gerade bearbeitet. Dies ist besonders nützlich, wenn mehr als eine Partitur sichtbar ist.

Das Fenster Einstellungen Anzeigen enthält Einstellungen für die Anzeige.

Die Anzeige-Breite gibt die Breite in Pixel an. Voreingestellt ist die Breite des Schirms.

Schalten sie die Option "Zeige Mitspieler-Namen" aus, im Noten-Editor (Die Namen werden oben im Menü eingegeben).

Takte/Zeile gibt die Anzahl der Takte auf dem Bildschirm an. 1 ist die Voreinstellunge, aber auch 2, 3 und 4 sind gute Werte.

Der Positions-Modus zeigt die Reihenfolge an, in der die Blöcke angezeigt werden. Im Blockbasierten Modus, werden die Blöcke numerisch angeordnet: Block 0, 1, 2, usw. Im Linearen Modus, wird die Reihenfolge der Spielsequenz angenommen.

Wenn die Spielsequenz 002 003 002 000 001 ist, dann wird 2, 3, 2, 0 und dann 1 angezeigt. Die Blöcke werden also so angezeigt, wie sie gespielt werden. Das ist auch wichtig für den Ausdruck!

FastMemPlay

main

1.54 FastMemPlay

FastMemPlay ist eine neue Eigenschaft für den 4-Kanal-Modus. Wie der Name schon sagt, werden die Samples vom FastMem aus gespielt, statt wie bisher aus dem Chip-Ram. Daher ist es jetzt möglich den gesamten Speicher zu benutzen. Es bietet auch einige andere Vorteile. Die Samples können beliebig lang sein (bisher gab es eine Beschränkung auf 131072 Bytes) und sie können auch ungerade Längen für die Wiederholung benutzen (diese Fähigkeiten sind automatisch verfügbar, wenn man die neue Misch-Routine auswählt).

FastMemPlay Fenster

ARexx Befehle Triggern

main

1.55 ARexx Befehle Triggern

Der neue OctaMED Befehl 2D wird benutzt, um Events zu kontrollieren, die mit dem Lied synchronisiert werden müssen. Sie können bis zu 256 Events definieren, die mit dem Befehl 2D gestartet werden können. Die möglichen Events sind: ARexx Befehl an OctaMED oder ein anderes Programm senden, das starten eines ARexx Scripts oder sogar eines anderen Programmes.

Noten ersetzen

main

1.56 Noten ersetzen

Dieses neue Fenster ermöglicht ihnen umfangreiche Such-Und-Ersetzen Funktionen auf Noten, Instrumente, Befehle und Kombinationen davon anzusetzen. Beispiel:

```
    C-2 2xxxx -> D-2 3xxxx
    xxxxx0FFF -> xxxxx0000 (entferne Befehle 0FFF)
    xxxxx09xx -> xxxxx0000 (entferne alle TPL Befehle)
```

Die Funktionalität dieser Eigenschaft wird erst duch ARexx-Skripte ersichtlich.

MIDI Kontroller

main

1.57 MIDI Kontroller

Das Set von Befehlen 31 - 3F wurde nun für MIDI Benutzer reserviert. Die alte Methode war, die Kombination von 05 und 00.
Sie können jetzt auch beliebige Kontroller auf die einzelnen Befehle 31 - 3F legen. Darüberhinaus können sie MIDI registrierte und nicht-registrierte Parameter-Zahlen mit dem gleichen Befehlssatz kontrollieren (3 Paare von 05/00 Befehlen wären dazu nötig).

MIDI Slave Modus

Verbessertes ARexx Interface

main

1.58 Verbessertes ARexx Interface

Es gibt 28 neue ARexx Befehle und 9 verbesserte Befehle. Einige neue Befehle umfassen das Finden und oder Ersetzen von von Noten/Instrumeten/Befehlen (siehe Replace Notes), das direkt Handling von Puffer-Inhalten und das Senden von MIDI-Meldungen (nützlich für Tastatur-Kürzel).

Die obigen Eigenschaften sind die wichtigsten Änderungen seit V6, aber nicht alle. Ich hoffe, das das Jahr Arbeit, das drin steckt, Sinn macht, und es ihnen gefällt.

Kleinere Eigenschaften

main

1.59 Kleinere Eigenschaften

* Ein einfaches Fester, um Instrumente anzuschauen und auszuwählen .
* Standard Verzeichnisse für Lieder, Instrumente und ARexx Scripts können gesetzt und gespeichert werden.
* ARexx Skripts können jetzt über einen neuen Menüpunkt gewählt werden.
* Statt das Dateiformat durch einen Requester auszuwählen, gibt es jetzt für jedes Format einen eigenen Menüpunkt.
* Das Lied-Kommentar-Fenster kann jetzt (optional) automatisch geöffnet werden, sobald das Lied geladen wird.
* Schneiden/Kopieren betrifft jetzt optional entweder alle Befehlsseiten eines Blocks oder die aktuelle.
* Ein spezieller Slave Modues wurde eingerichtet um den Amiga als MIDI Slave zu benutzen (z.B. zum Abspielen).
* Veränderte Einstellungen werden in der Titelleiste angezeigt (optional).
* Das Allgemeine Gleiten ist jetzt schlauer. Es kann vorzeichenbehaftetes Gleiten durchführen und spezielles Gleiten benutzen, wobei beide Ziffern unterschiedlich gehandhabt werden.
* Der Befehl 0FF7 beendet das Abspielen, bis alle vorhandenen SysEx Befehle gesendet sind.
* AmigaDOS Shell
  Öffnet eine AmigaDOS Shell, so wie auf der Workbench.
* Die Audio-Kanäle werden nun erst allokiert, wenn sie zum ersten Mal benutzt werden.
  Wenn sie den Mischmodus zusammen mit der Toccata, MaestroPro oder Delfina benutzen, dann werden sie überhaupt nicht benutzt!

-Teijo Kinnunen

A Quick OverView

main

1.60 A Quick OverView

Now For A Quick Overview Of What The New Additions Do......

These are the most important new features to be added, so far.
(With thanks to Ed Wiles, our Author who has been quite unwell recently)

The Mixing Parameters Window

Probably the Soundstudio's biggest addition is a new channel mode: Mix. Based on the mixing technique used in 5 to 8 channel mode, it can play 64 notes at once using the normal Amiga sound capabilities! You can also bring your song to life with effects such as echo, and use it to record part of your song direct to disk as a sample.

But before you get too excited, remember that the Amiga physically only has four sound channels, so Mix mode uses very special tricks to cram all those notes in and in the process, the notes can lose sound quality on slow processor
.
In fact, the faster your processor, the better quality the notes can be. For acceptable quality, you really need a minimum 68020 processor found in the A1200.

The Mixing Parameters window is used to set up Mix mode. But before I explain it, I'll list the enhancements and limitations that Mix mode has, in comparison with 4 channel mode.

Other Enhancements at this time

(1) Can play up to 64 notes at once.

(2) Effects: Echo, Cross Echo, Stereo Separation.

(3) Track Panning: Can alter the stereo location of each track.

(4) Many additional player commands.

(5) Sample length limited only by available memory
(previous limit was 131072 bytes).

(6) Direct-to-disk recording.

(7) Three new octaves: 2 low, 1 high. So the note range is now C-1 to B-6.
The 4-channel notes C-1 to B-3 are now notes C-3 to B-5.

(8) More precise sample loop setting. Previously, only even values of Repeat and RepLen (Instrument Parameters were possible.
Now, all values are allowed.

Limitations

(1) A probable loss of sound quality on old processors, but the faster your processor, the better the quality.
In fact, very fast processors will increase sound quality.

(2) Aura 16-bit samples can't be used

(3) Multi-octave samples can't be used.

(4) The oscilloscope equalizer doesn't function.

So although Mix mode is based on the old 5 to 8 channel mode, you will see that it has hardly any of its previous limitations.

Mixing Mode Section

main

1.61 Misch-Modi

Das wichtige Misch-Parameter Fenster wird nun beschrieben

Die Radio-Buttons links oben bestimmen das Ausgabemedium.

Amiga 8-bit, Amiga 14-bit

Diese Modi benutzen den eingebauten Audio-Chip Paula und deshalb funktionieren die auch mit jedem Amiga. Der 8-Bit Modus ist lauter als der 14-Bit Mouds, aber auch rauschiger. Der "pseudo" 14-Bit Modus benutzt eine Kombination von 2 Amiga-Kanälen um einen "14-Bit" Kanal zu erzeugen, der einen bessere Dynamik besitzt.

Die Mischfrequenz kann relativ frei gewählt werden. Die Obergrenze hängt vom Bildschirmmodus ab. OctaMED wählt die Obergrenze anhand des Bildschirmmodus. Wenn Sie den Bildschirm ändern, ändert sich die Frequenz.

Der Mischpuffer kann eine Größe zwischen 32 - 32767 annehmen, auch wenn es manchmal gerundet wird. Für MIDI sollte ein niedriger Wert gewählt werden.

Toccata 16-bit

Dieser Modus benutzt die Toccata-Karte von MacroSystems für hoch-qualitative 16-Bit Ausgabe. Die Unterstützung geschieht durch die toccata.library die jeder Toccata beiliegt. Es kann jede Version benutzt werden, aber V6 gibt bessere Resultate und das System läuft stabiler.

Die aktuelle Toccate besitzt 14 verschiedene Ausgangsfrequenzen.

Die Größe des Mischpuffers wird immer auf 512 Samples gerundet. Die Toccata kann zusammen mit MIDI benutzt werden, aber benutzen Sie hohe Mischfrequenzen mit kleinen Puffern (512 oder 1024).

Wenn Sie Probleme haben sollten (z.B. Unterbrechungen oder Stops), probieren Sie mal die Puffergröße zu erhöhen, oder die Mischfrequenz zu erniedrigen... Sie können alles zwischen 512 und 32000 eingeben.

MaestroPro 16-bit

MaestroPro ist eine digitale 16-bit Soundkarte von MacroSystems. Da MacroSystems keine Support Library für die MaestroPro liefert, haben wir die maestix.library von Richard Körber als Basis genommen. Diese ist frei kopierbar und befindet sich z.B. auf dem Aminet und zusammen mit OctaMED.

Es gibt drei verschiedene Ausgabe-Frequenzen (32 kHz, 44.1 kHz und 48 kHz). Sie müssen ein Referenzsignal erzeugen und die entsprechende Mischfrequenz in OctaMED auswählen. Wenn kein Referenzsignal ausgewählt wird, wird die interne 48 kHz Clock benutzt.

Die Größe des Mischpuffers wird auf 1024 Samples gerundet. Normalerweise benutzt man große Mischpuffer (ca. 32000 Samples oder so). Dies macht MaestroPro ungeeignet für MIDI und Echtzeit-Bearbeitung (aufgrund der langen Antwortzeiten des Puffers).

Disk 8-bit, Disk 16-bit

Diese Ausgabe-Modi werden benutzt um ein digitales "Image" eines Liedes auf Disk zu erzeugen. Dieses "Image" ist eine Sample-Datei, die benutzt werden kann, um Schlagzeugloops zu erzeugen, die sonst mehrere Spuren benötigen würden. Oder sie können das Lied von HD sehr resourcenfreundlich von HD abspielen. Oder sie könnten es benutzen, um es dann auf CD zu pressen. Die Smooth-Funktion ist besonders nützlich, da sie die Qualität erheblich verbessert.

Es gibt verschiedene Ausgabeformate. Diese hängen von der Ausgabeauflösung (8-Bit oder 16-Bit) und ob das Lied Mono oder Stereo ist, ab.

Die Ausgabe-Frequenz kann frei gewählt werden. Die Aktuelle Frequenz ist dabei immer gleich wie die gewünschte.

Der Mischpuffer kann frei gewählt werden und hat keine Auswirkung auf das Ergebnis, allerdings erhöht ein großer Puffer die Geschwindigkeit.

Siehe

Das Aufnahme-Fenster

.

Die Ausgabe ist Mono, ausser man schaltet explizit den Stereo Modus ein. Mono ist schneller als Stereo.

Smoothing erhöht die Ausgabequalität, aber ist sehr langsam. Aufgrund seiner Geschwindigkeit wird es nur für die Modi Disk 8-Bit/16-Bit empfohlen, aber sie können es gerne in den anderen Modi ausprobieren...

Delfina 16-bit

Dieser Modus benutzt den D/A-Konverter der Delphina-Audiokarte für hoch-qualitative 16-Bit. Es benutzt keine DSP Eigenschaften der Karte, sondern einfach nur den Ausgabe-Teil. Die Delphina Systemsoftware (delfina.library) wird benutzt. Teil des Supportcodes und das Testing wurden von Teemu Suikki gemacht (danke!).

Delfina besitzt 6 Ausgabe-Frequenzen (ab V2).

Der Mischpuffer kann eine Größe von 32 - 4096 samples besitzen.

Die Mischfrequenzen

main

1.62 The Mixing Frequency

The mixing frequency is a very important value.
It specifies, in Hertz (sample values per second), how quickly the samples should be mixed.
The higher the frequency, the better the sound quality, but also the more work the processor has to do.

So - you've guessed it - the highest mixing frequency that you can use depends on your processor. If you try using a frequency that is too high for your computer, the computer will hang: the mouse pointer will move very slowly, or even not at all. If this happens, do a panic stop by holding down both mouse buttons for several seconds.

Believe it or not, the highest possible frequency also depends on the screen mode! If the screen is DblPAL, DblNTSC or Productivity, the Mixing Frequency slider can take any value.
In other screen modes, the slider's maximum value is 28375.

If the output device can't play at your requested frequency, OctaMED chooses the frequency closest to your request.
That request is shown in the Requested numeric box, the actual frequency used in the Actual box.
The reason for this discrepancy is that all of the output devices, excluding Disk 8-bit/16-bit, can only play at certain frequencies.
(This even includes your Amiga...)

Some Technical notes:

The frequency of each note is different in 4-channel mode from Mix mode.
This is because in 4-channel mode, the frequencies are approximated to what the Amiga DMA can play.

Mix mode doesn't use the Amiga DMA, so it uses the correct frequencies.

The frequencies are very slightly off on Amigas not having an FPU and this is because the integer version of the frequency calculation has some error. The difference however is not audible.

Other Gadgets

main

1.63 Other Gadgets

* Max. Channels: The maximum number of channels. For example, if you want to play notes on tracks 0 to 5, set this slider to 6. Notes become quieter as you increase Max. Channels, to make room for the new channels. This slider doesn't increase processor load by itself: the load depends on the number of notes actually playing.

* Volume Adjust: Allows you to adjust the overall volume of notes. The value is a percentage; when 100 %, the notes are at normal volume. Set the notes to half-volume by sliding to 50%; double volume is 200%.

  In practice, increasing the volume above 100% generally causes unwanted distortion (noise), unless the samples in your song are quiet. Also, using effects can cause distortion, even at 100%. If this is the case, turn the volume down until the distortion is removed. (You can't hear the volume adjustment until you release the slider button.)

* Mix Buffer Size: The size of the mix buffer, an area of memory used by OctaMED to mix samples together. You needn't ever change it, unless:

  (a) in Disk 8-bit/16-bit mode. Increasing Mix Buffer Size to, say, 30000 greatly quickens direct-to-disk recording.

  (b) using MIDI. MIDI notes are played immediately, while mixed notes are played after a slight delay. So it's recommended to set Mix Buffer Size to the lowest allowed value, 32. If sound quality suffers as a result, compensate by raising the mixing frequency (if possible).

  (c) you use samples with very short loops. Playing a loop shorter than about a third of the Mix Buffer Size (i.e. usually 100 bytes) considerably increases processor load. So if you must use very short loops, try decreasing Mix Buffer Size.

* Panning and Effects: Open the Mix - Track Panning and Mix - Effects windows. They allow you to adjust the stereo location of each track, and add effects such as echo to your music.

Other Points Of Interest

main

1.64 Other Points Of Interest

(1) 5 - 8 channel mode uses a frequency of 15768 Hz in non-HQ mode ↩ , and 28867 Hz in HQ mode. As a side note, you could *perfectly* reproduce the 4-channel mode with a mixing frequency of 3.6 MHz... :-)... Though 48 kHz with smoothing is often better in practice.

(2) With 16-bit samples, only volume values 0, 1, 2, 4, 8, 16, 32 and 64 can be used. This means that note volumes may sound a bit strange if you change Volume Adjust or the track panning. Also, the volumes you can use with some player commands (such as Set Volume, type 0C) are limited to the above values.

(3) To allow old 4-channel and 5 to 8-channel songs to use the new Mix mode, the Miscellaneous Options window contains two new Use Mixing check boxes. When on, the samples in loaded songs are transposed up 2 octaves. (This doesn’t include synthsounds, MIDI instruments or ExtSamples.)

When using old 5 to 8-channel songs with the new mode, don’t try to set Volume Adjust to 200 % to allow for halved samples. Halved samples are in 7-bit quality, so it’s best to re-load the original samples.

(4) Internally, OctaMED has three separate parts which control playing, called the players. There’s one player for 4-channel mode, another for 5 to 8-channel modes, and another for Mix mode.

When you play a note using the keyboard in 4 or 5 to 8-channel modes, the player is used briefly to start the note off, but then the note is left to play and finish of its own accord. The Mix mode player, however, is used throughout the note’s playing, from its start to its finish.

So the Mix mode player is switched on when the first note is started. But it isn’t switched off when the note has finished. Instead, to react more quickly to other notes you play, it’s kept switched on until you click STOP or press the space bar.

This means that the player is switched on even when no notes are being played. On slower Amigas, this may slow down operation, so simply press the space bar to switch the player off. It also means that if you change anything in the Mixing Parameters window, such as Mixing Frequency or track panning, you must stop playing and restart it for the changes to take effect. (This doesn’t include Stereo Separation, Echo Depth, or a change from Echo to Cross Echo or vice-versa.)

(5) Do remember about panic stop! If the computer seems to lock up during play, try holding both mouse buttons down for a few seconds.

(6) 5 to 8-channel mode’s split channels don’t exist! Amiga volume registers are set to a fixed volume. The volume is scaled by the mixing routine, so each track does have an independent volume.

(7) Almost all settings in Mixing Parameters, Mix - Track Panning, and

Mix - Effect are saved with songs. In particular, Mixing Mode and Mixing Frequency aren't saved. This is so that, for example, a song created on a fast Amiga using a high frequency won't immediately cause a slow Amiga to lock up.

OctaMED uses a special file format for songs using Mix mode: MMD3. This is identical to MMD2. The only reason for its existance is so that older versions/player programs won't attempt to play Mix-mode songs.

(8) You can play samples in either type of memory - Chip or Fast - in Mix mode. By default, though, samples are loaded into Fast memory, because Instr menu - Load Samples To Fast Mem is automatically switched on.

Other Gadgets

Track Panning Window

main

1.65 Track Panning Window

Here you adjust the stereo location of the notes played on each individual track, do this using the sliders.

Track numbers are displayed to the left of each slider, the tracks' stereo location to the right. To begin with, all tracks have a stereo location of 0 (center). This means that notes on all tracks are played with equal volume on both speakers; in other words, they're played in mono.

To force a track's notes to be played entirely through the left speaker, drag the track's slider all the way to the left (value -16). Likewise for the right speaker: drag the slider to the far right (value 16). Intermediate values play the notes at different volumes on each speaker.

Free Panning and Sum Of Balances

Because track panning is really altering the volume of each track on each speaker, you must be careful that the volume on either speaker doesn't become too high, causing distortion. For example, setting all tracks to be played on the left speaker (value -16) is bound to cause distortion on the left speaker.

The Sum Of Balances display helps you prevent distortion. It displays all the tracks' stereo locations added together. When the volume is perfectly balanced between the two speakers, the Sum Of Balances is 0, so adjust the sliders until it becomes 0.

Free Panning, when on, allows you to set the stereo locations to whichever values you choose, without worrying about volume distortion. Not surprisingly, Free Panning is usually on.

Special Effects Window

main

1.66 Special Effects Window

Add special effects to your music.

The one you'll be dying to play with is echo. To switch echo on, set the Echo cycle gadget to Echo or Cross Echo. The difference between the two is Cross Echo alternates echoes between the speakers; you must have Stereo mode on to use it.

Echo Rate is the distance, in milliseconds, between each echo.
It can take any value in the range 1 to 32767.

Echo Depth sets the depth of echoing. The larger the value, the deeper the echoes. Technically, it specifies the relative amplitudes of successive echoes. For example, if it's 25 %, the first echo's amplitude is 25 % of the original amplitude, the second echo is 25 % * 25 % = 6.25 % of that.

Stereo Separation is interesting. Dragging the slider to the right separates the sound on each speaker. Dragging to the left brings the speakers' sound closer together. This feature is best understood by experimentation. (Technically, the stereo image is separated by feeding part of the left channel to the right in inversed phase, and vice versa.)

The Recording Window

main

1.67 The Recording Window

This window is displayed during direct-to-disk recording, the act ↩
of
transferring part of your song to disk as an actual sample.

To record directly to disk:

(1) In the Mixing Parameters window, select Disk 8-bit or Disk 16-bit,
depending on whether you'd like an 8-bit or 16-bit sample.
Also set the Mixing Frequency to any value you wish.
(the higher the frequency, the larger the produced sample).

(2) The next time you play a note or your song, a Record as file
requester will appear. In this requester, select the filename of the
produced sample.

(3) Next, you set the file type of your sample in the requester that appears.

(4) Finally, the Record window will open and OctaMED will start recording. The window displays information on the sample's file format (resolution (8/16-bit), IFF/RAW/MAUD etc, Mono/Stereo, frequency). It also displays recording time in minutes and seconds, and file size. These two values change as the song is being played.

Click the Stop Recording button to stop recording. You must do this because OctaMED doesn't stop recording automatically at the end of the song or note. You can now load your sample back into memory.

MIDI Slave Mode

main

1.68 MIDI Slave Mode

Using the MIDI menu's Slave Mode Active item, you can turn your ↩ Amiga into
a MIDI device!
Why on earth would you want to do that?
Well, if you're lucky enough to own two Amigas, you can use the sound channels of both Amigas together, to play a total of 8 samples at once rather than the usual 4 and you do this by using one Amiga to control the other, as if it were itself a MIDI keyboard.

It works best in 4-channel mode, for highest quality. You compose your song using one of the Amigas (call it the master), and use the other Amiga (the slave) only as a note player. The blocks in your song should be 8 tracks wide: use tracks 0 - 3 for instruments played by the master Amiga, and tracks 4 - 7 for those played by the slave Amiga.

The idea is, the slave Amiga's first 16 instruments (01 - 0G) correspond to the 16 MIDI channels. So when the slave Amiga receives a MIDI message to play a note on MIDI channel 4, it plays the note using instrument number 04. One slight drawback is each instrument can only be played using one particular sound channel, which you designate using the master Amiga.

Anyway, here's how you set it all up:

(1) Attach a MIDI interface to each of the two Amigas. Connect MIDI OUT on the master Amiga to MIDI IN on the slave Amiga.

(2) Load OctaMED into both Amigas. Select MIDI menu - MIDI Active on the master Amiga. Select MIDI Active, Input Active and Slave Mode Active (all in the MIDI menu) on the slave Amiga. Make sure both Amigas are in 4-channel mode.

(3) Now to set up the instruments. Go to each Amiga in turn, and load the instruments you want to be played on that Amiga. On the slave Amiga,

you're restricted to instrument slots 01 - 0G; on the master Amiga, you can use any slots.

(4) On the master Amiga, you actually need two categories of instrument: those to be played through the master Amiga (which you've just loaded), and MIDI instruments that correspond to each instrument you've loaded into the slave Amiga. OK, let's say you've loaded a sample named Fantasia into slot 06 on the slave Amiga. So on the master Amiga,

a) Select any empty instrument slot

b) Open Instrument Parameters

c) Change the instrument Name to Fantasia [slave] (for example)

d) Slide the MIDICh slider to 6 (because Fantasia is loaded into slot 06 on the slave Amiga)

e) You must also tell OctaMED which sound channel on the slave Amiga that Fantasia should be played through, using the Preset slider. So if Fantasia is to be played through channel 2, slide Preset to 2. Because there are 4 sound channels, each played using one of tracks 0 - 3, you can slide Preset to 0, 1, 2 or 3.

Be careful when deciding which instrument should be played through which channel. If you've loaded up to 4 instruments into the slave Amiga, you just need to assign a different sound channel to each instrument. If you have more than 4 instruments, though, you should take care that no two instruments are set to play through the same channel at the same time, because this is impossible! This depends on your song.

f) Close Instrument Parameters

Now when you want to play, say, note G-2 using the Fantasia instrument, you should enter a G-2 using the Fantasia [slave] instrument on the master Amiga. Repeat steps a to f for each instrument you've loaded into the slave Amiga.

(5) Now compose your song! Use 8-track blocks. For instruments played through the master Amiga, use tracks 0 - 3 as usual. For those played through the slave Amiga, use tracks 4 - 7 and the MIDI instruments you've set up to correspond with the slave Amiga's instruments. Then just play the song: it should be quite an amazing effect!

Other notes:

a) Slave mode only receives and handles MIDI Note On messages. So any effects received, including player commands, will be ignored.

b) You can, however, use player command 0C (set volume) with notes played by the slave Amiga. The instrument's default volume (in Instrument Parameters) changes to the appropriate level when a 0C is received. The Instrument Parameters slider isn't updated, though, for effciency reasons.

c) To reduce the amount of MIDI data sent, you should really switch on

Suppress NoteOff on all of the MIDI instruments defined on the master Amiga.

The MIDI Cmd 3cxx Window

main

1.69 The MIDI Cmd 3cxx Window

As its title suggests, this is one of OctaMED’s more technical windows!

Using a combination of player command types 05 and 00, you can set any MIDI controller you choose. Command 05’s level is the controller number, command 00’s the controller value.
So if OctaMED came across this in a song: 00509 --- 00004 then OctaMED would set MIDI controller number 9 to 4.
The obvious disadvantage with this is that it takes two lines:
one to provide the controller number, another to provide the value.

This window cuts the setting of MIDI controllers down to one line.
You define command types 31 - 3F to set your chosen MIDI controller numbers to the value given by the command level you use in the song.

For example, you can define player command type 35 to set MIDI controller number 12. Then, if you use command 3506 in your song, OctaMED sets MIDI controller number 12 to 6. So, only one player command is needed to set MIDI controllers, instead of two.

The gadgets

Starting at the top, the slider selects a player command type.
A value of 5, for example, selects type 35. Types 31 to 3F can be selected.

Clear Setting clears the selected player command type’s setting. Its controller type is set to Standard [MSB], and its number to 0. Clear All does this to all player command types.

The other gadgets show the selected command type’s setting: its controller type and number. The controller type can be Standard, RPN or NRPN, together with [MSB] or [LSB].

More information as and when further developed.

You can make a copy of the current shortcut by holding down Shift while clicking Ins. New or App. New. This is useful for making a new shortcut similar to the current one.

You can now create an empty shortcut for the Help key, thus turning the press Help for Help feature off - the Project menu item can be selected instead. This avoids situations where the rather large Help file is accidentally loaded instead of deleting a note (i.e. Help rather than

Del is pressed).

Playing PC Mods

main

1.70 Playing PC Mods

In preperation for the inevitable release of the PC version of our program
OctaMED can now load modules created on two different PC sequencers:

ScreamTracker 3 (S3M) and FastTracker 1.0. (see below)
Not all S3M effects are supported, but most modules play without trouble.
The effects that are supported are:

Extra fine slides (commands EEx, EFx); Tremor (command Ixy);
Retrig (+ volume-slide) (command Qxy)

At time of writing the more popular FastTracker2 is not *yet* supported.

FastMemPlay Window

main

1.71 FastMemPlay Window

This poetically-named window allows you to play samples from Fast memory.

By editing the Buffer Size box, you can alter the FastMemPlay buffer size.
A small value, such as 32 or 64, is recommended to minimise distortion if
your song contains synthetic sounds.

Load Samples To Fast Mem

There are two types of memory: Chip memory and Fast memory.
Fast memory is much faster than Chip memory, so it's preferable to store
things in there. However, the Amiga usually needs all samples to be stored
in Chip memory.

With OctaMED, you can store samples in either
type of memory. However, this depends on the
song's channel mode Samples must be in Chip
memory if the song is in:

(1) 4-channel mode
(2) 5 to 8-channel mode, and contains

non-paired tracks.

But even in these modes, samples can be in Fast memory if you use the FastMemPlay facility. All a bit confusing!

Anyway, these 3 menu items allow you to store samples in your chosen type of memory. When Load Samples To Fast Mem is on, any samples you load are stored in Fast memory. When off, they're stored in Chip memory as usual.

Load Samples To Fast Mem is automatically turned on when Mix mode is selected, to take advantage of the increased speed. So to load samples into Chip memory in Mixing mode, switch this item off.

Samples cannot be shared between the two types of memory: they must all be in Chip, or all be in Fast. So if you decide to store samples in Fast memory, any samples currently in Chip memory must be moved to Fast memory. The Move Samples To Fast/Chip Mem exist for this purpose.

Replace Notes

Opens the Replace Notes window Here, OctaMED replaces all Source notes found in a particular area of the song with the Destination note.

Player commands interpret their command level in one of three different ways, depending on the type. Some, like 0C (set volume), take both digits together as a single positive number. Others, like 04 (vibrato), take each digit separately as different numbers. With vibrato, the 1st command level digit represents speed, the 2nd depth.

Still others, like MIDI command 03, take both digits together as a signed number. So command 03's level can range from -128 to 127 (decimal), instead of the normal 0 to 255.

Previously, Generic Slide always assumed the command level to be a single positive number. Now it's more intelligent!

Slave Mode Active

When on, your beloved computer becomes a MIDI device, slave to whatever is controlling it! See MIDI Slave Mode for more information.

Immediate Preset Change

When a particular instrument's Preset value is changed (Instrument Parameters usually a preset change message is sent the next time a note is played with that instrument.
With this item on, a preset change message is sent straight away.

FastMemPlay

Raw Sample Conversion

main

1.72 Raw Sample Conversion

If you load a RAW sample that sounds very ↩ distorted and
noisy, try using this submenu to correct it.
OctaMED's samples are stored in 'signed' format.
Most samples originating on PCs are 'unsigned',
so use Unsigned <- Signed to convert them.

Use Swap Byte Order on noisy 16-bit samples. Each value of a 16-bit sample takes 2 bytes of memory. Swap Byte Order swaps the order of each value's bytes. This can solve distortion problems.

ARexx Trigger Setup

main

1.73 ARexx Trigger Setup

The ARexx Trigger Setup Window

This window gives player commands the keyboard shortcut treatment.
Just like with keyboard shortcuts, you can now set player commands (of type 2D) to execute an OctaMED ARexx command or ARexx script, send an ARexx command to another program, or load and run a program from disk.

At the top of the window, select a command level to edit using the slider. For example, to edit command 2D45, set the slider to 45. Clear Current sets the current command's action to None. Clear All does the same to all commands.

The Action area describes what the player command is to do if it appears in a song. It is exactly like the Action area in the Keyboard Shortcuts window (and also in the Input Map Editor).

OctaMED Command

Executes an OctaMED ARexx command (type it into the Command box, together with any required parameters). ARexx itself isn't actually involved. To execute more than one command, use the OP_MULTICMD command.

(See manual for more information on OctaMED commands)

Execute ARexx File

Executes an external ARexx file (type its name into the command box). Unless the file is in PROGDIR: or REXX:, specify the full path name.
The filename should end in .omed

Ext. ARexx Command

Sends an ARexx command to another program. Type the command into Command, and the program's ARexx port name into ARexx Port.

Note: To use Execute ARexx File or Ext. ARexx Command successfully, make sure you have run the program RexxMast.

Launch Program

Launches (runs) an executable program file.
Type its name into Command.

The 1-64 Ch Mixing sets the new Mix mode.
With only a slight loss of sound quality, this gives you up to 64 channels, special effects, track panning and more.
You can also use MIDI in Mix mode.

Other Features

main

1.74 Other Features

Annotation Window

Use this window to attach any text to your song. The text might be a copyright notice, the author's name, explanatory text, or best of all those famous greetings which will no longer be lost.

In the text box at the very top of the window, you may type anything up to 70 characters long. When the song is reloaded, the contents of the text box will be displayed on the screen's title bar.

In the remainder of the window, you can attach any text file to a song. Create the file in an external text editor, then click Load File to load it. Save Text saves the text under a chosen name, and Discard Text removes the text from memory. The text is saved with songs.

Note that tab characters (ASCII code 9) can’t be used in the text.

Show After Loading affects what happens when a song with annotation text is loaded. Normally, the Song Annotation window is immediately opened, displaying the song’s text. Switch Show After Loading off to prevent this.

The Instruments Window

This window displays a list of all your instruments.
You may select one from the list, to make it the current instrument.

The list displays all instrument numbers and names. Click on an instrument to make it current. So this is an alternative selection method to using Shift-<left and Shift-<right, or using the Instrument Parameters window.

> Loop Ping-Pong is only used in Mixing mode. When on, OctaMED plays the sample’s loop (if it exists) in a different way: forwards and backwards alternately.
>
> In other words, the sample is played until the end of its looped part; the looped part is played *backwards* from its end to its start; the looped part is then played forwards from its start to its end; then backwards; then forwards; and so on.

The Replace Notes Window

Here, you give OctaMED a source note and a destination note. Then OctaMED replaces all source notes found in a particular area of the song with the destination note. For example, you can replace all C-2 10C32’s in the current track with C-2 10C48 if you like.

Setting the source and destination notes

Set the source and destination notes in the Source and Dest boxes, both initially --- 00000. As you might have guessed from the five 0’s, you also set the note’s instrument number and player command.

You set the Source and Dest boxes in exactly the same way as in the Programmable Keys window. To set a note, hold down the left mouse button over the note in the box (initially ---), then press the note’s key and release the mouse button.

For example, to set the Source note to G-2: Position the mouse over the note in the Source box. Now hold down the left mouse button, and press T. Assuming that Oct is set to 12 (Main Control window), the note becomes G-2. To set the note to ---, use the Del key.

To set an instrument digit or player command digit, hold the left mouse button over the digit, then type the digit and release the mouse button. Quite easy really.

Transparent notes/digits

When setting a note or digit, try pressing Return instead of pressing the note’s key or typing the digit. The note or digit is now transparent, and is shown as ’x’.

Transparent notes or digits are ignored by OctaMED. So if the Source box contains xxx x0C32, OctaMED will replace all 0C32 player commands, regardless of their note and instrument number. Similarly, if the Source box contains G-3 4xxxx, then OctaMED will replace all G-3's played by instrument 04, regardless of what their command digits are.

Transparency in the Dest box has a slightly different meaning. If the Dest box contains xxx x0C32, then the player command of all Source notes will become 0C32: the note and instrument number will remain unchanged. Similarly, if the Dest box contains G-3 4xxxx, all Source notes will become G-3 and will be played by instrument 04: their command digits will remain unaffected.

Replacing the notes

In the lower half of the window, the six buttons replace the given notes in one of six different areas. These areas are as in the Transpose window, aside from Note which replaces just the Tracker editor's current note.

For example, clicking Block replaces the Source note with the Dest note throughout the current block. Sel. Tracks replaces throughout the selected tracks (a track is selected by clicking on its S gadget at the top of the Tracker editor).

Transparency and the choice of six areas make this window pretty powerful.

Swap Amiga Channels (Mix)

Sound channels 0 and 3 should be played through your left speaker
and 1 and 2 through your right.
On some Amigas, these are swapped round!
So switch this check box on to correct them.
It only has an effect in Mix mode.

Use Mixing (MMD) or Mixing (Tracker mods)

Makes the necessary changes to MMD (normal) songs and Tracker songs
to use Mix mode.
This involves transposing all instruments, except synthetic
and MIDI, up two octaves.

Thats all for now, we await Ed's belated work with eagerness!!!!

-Ray

RBF Software. May 96

Soundstudios New Features

ARexx Introduction

main

Go back to the very beginning of this guide

1.75 ARexx support

AREXX SUPPORT

ARexx Introduction

ARexx Features

Intro to Commands

First ARexx script

Writing a Script

1.76 Introduction

ARexx is one of these 'buzzwords' used by so many Amiga people, ↩ but many users seem to be in the dark as to what it actually is.
Well, in relation to OctaMED, it is both a way of adding new features to the program, and a time-saving alternative to many mundane tasks.

You use ARexx by typing in an ARexx script, a set of commands (instructions) that ARexx is to carry out. If ARexx were to control a toaster, some of these commands could be start toasting, cancel toasting, set timer, switch on "frozen bread" indicator and so on. But we'll be using ARexx to control OctaMED, so commands are more likely to be play song, open the Font window, or mark a range over track 3.

Well, it would be nice if commands were so informal, but computers find informal English very difficult to understand!
So commands are actually a lot more precise.
For example, the instruction "set instrument 04's default volume to 32" would in fact be performed by the two commands:
IN_SELECT 4 and IN_SETVOLUME 32.

Understanding how to use OctaMED's commands is vital for writing ARexx scripts, so let's look at them now:
have OctaMED loaded and ready because it's time for a tutorial!
(Admittedly it's a very quick tour but hopefully you'll pick up the basics).

Intro to OctaMED Commands

main

1.77 Intro to OctaMED Commands

First, simple commands and command groups

Close all windows but the main three (Main Control, Tracker editor, Information). Now select Project menu -> Command Shell. The window that appears is used for entering OctaMED commands.

If you enter a command that OctaMED doesn't recognize, or you make a typing mistake, you'll most likely get error code 25. Try this:

(1) Type: fix the car (and press Return)

Not in OctaMED's vocabulary. Pity really. But here's one that is:

(2) Type: ve_octamed (and press Return: do so after each command you type)
[To type a '_', hold Shift and press the key beside 0 (zero)]

OctaMED displays the command result, the version of OctaMED you're using. All OctaMED commands consist of a two-letter group name (VE here), an underscore (_), and the command itself (OCTAMED here). Group names reveal a lot about a command's purpose; for instance, we know immediately that VE_OCTAMED is a version command. Other group names are as follows:

Group name	Purpose	Group name	Purpose
ED	Edit	SA	Sample editor
IN	Instrument	SG	Song
OP	Options	SY	Synth editor
PL	Playing	WI	Window
RN	Range		

(By the way, this guide will tend to show commands in capitals. Do feel free to enter them in lower-case, though, it looks a bit nicer!)

Parameters and command templates

Armed with the knowledge that WI commands apply to windows, perhaps you can guess what the WI_OPEN command does.

(3) Type: wi_open songoptions

Yes, the Song Options window opens. The WI_OPEN command can't be used on its own; it needs a parameter (in this case, SONGOPTIONS). Parameters are extra pieces of information that many commands need to function properly. There can be more than one of them, and they're all separated by spaces.

Find WI_OPEN's template on page 66. It's WINDOW/A. This template describes WI_OPEN's parameter:

(a) WINDOW is its keyword;   (b) /A is its specifier

Specifier /A means that the WINDOW parameter is necessary; without it, WI_OPEN won't work properly.

(4) Close the Song Options window, then type: wi_open

Error code 10, because we didn't provide a window name. If you like, you can type the parameter's keyword (WINDOW in this case) before the parameter. This can serve as a reminder of what the parameter actually signifies.

(5) Type: wi_open window tempo

The Tempo window opens. We'll examine a new command now, SG_LOAD, and use it to load our demo song. Firstly we'll have a look at its template; and there's a quicker way than finding it in the manual...

(6) Click inside the OctaMED Command Shell window, then type: sg_load ?

A ? beside any command returns its template. Examine the template:

(a) NAME has no specifier: the parameter is a string. (A string is a series of any letters, numbers or symbols).

(b) FORCE has specifier /S: the parameter is a switch. Type the keyword to switch FORCE on; leave the keyword out to switch FORCE off.

(7) Type the following exactly:

```
      sg_load name octamed_soundstudio:Misery force  (press Return as usual)
```

If you managed all that, the demo song (Misery by Dave Sullivan) should load. Let's examine what we've just typed.
OctaMED_Soundstudio_CD:Misery is the NAME parameter, the name of the song.
We could have left the NAME keyword out, as we've seen, but we left it in to make things clear.

Including the keyword FORCE switches the "force loading" option on.
This loads the song irrespective of whether or not the current song has been modified in any way.
Omitting the FORCE keyword would switch the "force loading" option off.

So we've seen parameters with no specifier, with specifier /A, and with specifier /S. Let's look at the SG_SETTEMPO command (sets the song's tempo).

(8) Type: sg_settempo ?

The /N specifier means that the parameter is a number. There are three keywords (BPM=SPD, TPL and LPB), each of which changes a different tempo setting. In fact, BPM=SPD is not a single keyword: it means you can use either BPM or SPD as a keyword for that parameter.

(9) Type: sg_settempo spd 70

In the Tempo window, notice that the Tempo slider is now set to 70.
After entering these tempo-changing examples, feel free to use the commands PL_PLAYSONG and PL_STOP to try out the new playing speeds.
No, don't reach for that mouse, you're not allowed to!

(10) Type: sg_settempo bpm 57

Again, the Tempo slider changes (to 57 now). So the two keywords SPD and BPM are interchangeable for that parameter. Now let's try the TPL slider.

```
 (11) Type: sg_settempo tpl 12
```

The slider becomes 12. You can also change both Tempo and TPL sliders at the same time.

```
 (12) Type: sg_settempo tpl 8 spd 25
```

So far, we've been preceding every parameter with its keyword. A while ago, I mentioned that this is not strictly necessary but it makes things clear. Well, using keywords is in fact necessary if a command's parameters are entered in a different order from that given by its template.

For example, the template for SG_SETTEMPO is BPM=SPD/N,TPL/N,LPB/N. So entering the command SG_SETTEMPO 25 10 4 (without keywords) sets the Tempo slider to 25, the TPL slider to 10, and LPB to 4: the order given by the template. If the desired effect, however, was to set LPB to 25, Tempo to 10, and TPL to 4, then you'd need to use keywords.

This is also true for changing a single slider. SG_SETTEMPO 17 (without keyword) will always set the Tempo slider to 17, as this is first in the template. If you actually wanted to set the TPL slider to 17, you'd have to use SG_SETTEMPO TPL 17 instead.

Perhaps entering these examples will clarify a bit. Notice the changes in slider values after entering each command.

```
 (13) Type: sg_settempo 25 10 4
            sg_settempo lpb 25 bpm 10 tpl 4
            sg_settempo 17
            sg_settempo tpl 17
```

Another occasion where you must use the keyword is if specifier /K is given. ED_GOTO's parameters use this specifier.

```
 (14) Close the Tempo window, then type: ed_goto ?
                                        ed_goto block 4 track 2
```

Click on the Tracker editor's window depth gadget (far top right), and confirm that we're now in track 2 of block 4. Click the depth gadget again.

Did you understand ED_GOTO's template?
All of its parameters have specifiers /K and /N, so they're all numbers and they all require their keyword.

The last specifier we'll examine is /F.

```
 (15) Type: wi_showstring ?
            wi_showstring message I'm sorry, I can't fix cars.
```

Now close the OctaMED Command Shell window, and look at the title bar. /F means that all information typed after the keyword will count as that one parameter. So if the command has other parameters (unlike WI_SHOWSTRING), make sure they come before the one with /F.

Also, parameters containing spaces are usually not allowed, but they are allowed with /F parameters. (Other parameters need to be surrounded by double quotes. For example, "This parameter contains spaces").

There is one other rarely-used specifier: /M (multiple parameters). It means you can enter the same parameter over and over again: different notes of a chord, for example. OP_MULTICMD (see §12.9.6 number 5) uses /M.

OK, now that we have the basics of OctaMED commands under our belt, let's see how to string them all together by creating an ARexx script.

Our First ARexx Script

main

1.78 Our first ARexx script

Entering a script (using the Ed text editor)

Our first script will perform the simple (and admittedly not very useful) task of swapping track 0 with track 2.

You can't use OctaMED to enter an ARexx script: you need a text editor. Any text editor will do, and if you have a favourite by all means use it. I'll describe how to use Commodore's Ed text editor.

(16) Switch to the Workbench screen using the screen depth gadget (far right of title bar), or if you're a keyboard fanatic, press Left Amiga-N

(17) Click anywhere on the Workbench screen, then select Project menu -> Execute Command

(18) In the text box that appears, type the following exactly:
ed rexx:swaptracks.omed (and press Return)

Please make sure you're using a backup of your OctaMED disk!
Also make sure that this disk is write-enabled.
(Does not aply to CD users)

If all has gone well, you'll be presented with an empty Ed window.

(19) Now please type in the following script as accurately as you can, pressing Return at the end of every line. (' is on the key above Tab)

```
/* SwapTracks.omed: Swaps tracks 0 and 2 */

address octamed_rexx
options results

'ed_goto track 0'
'rn_copy track'

'ed_goto track 2'
```

```
    'rn_swap track'

    'ed_goto track 0'
    'rn_paste track'

    'wi_showstring Tracks 0 and 2 swapped.'
```

(20) When you've finished, select Project menu -> Save
Note that if using the CD, you will ned to save your ARexx scripts to your Hard Drive or a pre-formated floppy

OK. Let's have a closer look at this script.
The first three lines are very important:

i) /* ... */ A comment. You can type any text between the two symbols: a brief description of the script is a sensible move.

ii) address octamed_rexx Tells ARexx that it will be talking to OctaMED. OCTAMED_REXX is OctaMED's ARexx port.

iii) options results Necessary if OctaMED is to give ARexx information. Not really needed for this particular script, but best form habits.

These three lines, with perhaps a blank line in between lines 1 and 2 for neatness, must be at the start of every ARexx script you write for OctaMED. Yes, you can't even leave the comment out!

Now we progress to the main script. OctaMED is to move to track 0, select Track menu -> Copy, move to track 2, select Track menu -> Swap w/Buff, move to track 0, and finally select Track menu -> Paste. After all this, track 0 will have been swapped with track 2. Trust me!

Notice that all OctaMED commands are enclosed by single quotes ('). Make a habit of this, it's recommended for all commands that control programs. The quotes are actually necessary for some commands, and it also distinguishes OctaMED commands from other commands. Just use them, OK?

The script is rounded off nicely with a confirmation message that the tracks have been successfully swapped. Users like such messages.

(21) Quit Ed by clicking its close gadget, then return to OctaMED's screen using the Workbench screen's depth gadget

Executing a script (using the Keyboard Shortcuts window)

OK. From within OctaMED, the only way of executing (running) a script is by setting a keyboard shortcut for it. Our SwapTracks script will be executed by holding Shift and pressing F1.

(22) Select Settings menu -> Keyboard Shortcuts. In the left-hand section, click Ins. New. Now click inside the Name text box, delete <unnamed>, and type Shift-F1 (Swap tracks) (and press Return)

(23) In the middle (Input) section, click once on the Shift cycle gadget (it should now show Either). In the Raw box, change the 0 to 80.

```
 (24) In the right-hand (Action) section, click once on the cycle gadget
      (it should now show Execute ARexx File).
      In the Command box, type swaptracks (and press Return)

 (25) Save the shortcuts using Save (click OK in the requester),
      then close the window
      Remember that you cannot save to the CD version! :)
```

If you're unsure of the Keyboard Shortcuts window, re-read the Manual.guide

```
 (26) Watching the Tracker editor carefully, press Shift-F1
```

Notice tracks 0 and 2 swapping? With a nice message afterwards? Very clever. If you're a perfectionist, you can speed things up a bit by adding a couple of commands to the script. Do steps 16 to 18 on pages 55-56, then just after the options results line, type 'op_update off'. Then just before the wi_showstring line, type 'op_update on'. Save the script and quit Ed. Back in OctaMED, press Shift-F1 again and you should notice a difference.

You may have noticed that in step 24, we gave the script's filename as simply swaptracks (the full name is REXX:swaptracks.omed). This is possible as long as the file is in the REXX: directory and has the extension .omed.

Arexx Features

main

1.79 ARexx Features

We've seen how ARexx can be used to perform a series of OctaMED commands with a single keyboard shortcut, thus saving time. But ARexx can be far more powerful and time-saving than that, as I'll hint at in the next sections.

DO loops (how to repeat commands)

One time-saving ARexx feature is its ability to repeat the same set of commands as many times as you like. This device is called a DO loop.

We'll now enter another script, which will change every second note in track 3 to D-3.

```
 (27) Switch to the Workbench screen again, and select Project menu ->
      Execute Command. In the text box, type ed rexx:track3d3.omed.
      Now enter the following script:

     /* Track3D3.omed: Changes every 2nd note in track 3 to D-3 */

     address octamed_rexx
     options results

     'ed_goto track 3'
     do currline = 0 to 63 by 2
        'ed_goto line' currline
        'ed_setdata note 27'
     end

     'ed_goto line 0'
     'wi_showstring Notes changed.'
```

(28) Save and quit in one go by selecting Project menu -> Save & Exit
One final reminder, you will need to save to a floppy or
your HD if you are running this from the CD version.
Please remember this in all future dealings with the SAVE command

(29) Back in OctaMED, open the Keyboard Shortcuts window. Click Ins. New, and replace <unnamed> with Shift-F2 (Track 3 D-3). Cycle the Shift gadget to Either, and type 81 into Raw. Cycle the top-right gadget to Execute ARexx File, and type track3d3 into Command. Finally, save the shortcuts and close the window.

(30) Move to one of blocks 0 to 12 in our demo song (C-3 should be on every second line in track 3). Now press Shift-F2 to execute our new script.

This time it really is slow! So perhaps you'd like to add 'op_update' commands as mentioned earlier. (These commands 'switch off' the Tracker editor and Information window for the duration of the script).

Right. Let's puzzle over this script. After the three very important lines - remember? - the cursor moves to track 3. Now we have our DO loop.
The variable currline is set to represent the values 0 to 63 in steps of 2: 0, 2, 4, 6 and so on right up to 63. (Well, 62, 63 is an odd number).

What does this mean? Well, currline is firstly given the value 0. From then on, everywhere that ARexx sees 'currline', it replaces it with 0. So 'ED_GOTO LINE' currline (the first line of the DO loop) is replaced with 'ED_GOTO LINE 0'. So OctaMED moves to line 0.

The next line is ED_SETDATA NOTE 27 which sets note D-3 (I'll explain later). Then ARexx reaches the end of the DO loop, marked by END.
So it adds 2 to the value of currline (0), giving a new currline value of 2.

After checking that this new value is not more than 63, the DO loop starts again. 'ED_GOTO LINE' currline is replaced with 'ED_GOTO LINE 2'. OctaMED moves to line 2, and changes the note to D-3. ARexx reaches the end of the DO loop again, and adds 2 to currline, giving a new value of 4. And so on.

When currline eventually becomes 62, and the end of the DO loop is reached, ARexx adds 2 to currline as usual. This new value is 64, which is more than 63, so the DO loop finishes. ARexx continues with the line after END.

Out of all this come three main points:

i) The DO loop sets up its variable (currline in this case) for the whole loop straight after the word DO. Here, the variable runs from 0 to 63 in steps of 2. If it were to run in steps of 1, you would use
DO currline = 0 TO 63 BY 1, or more simply DO currline = 0 TO 63.

ii) Every occurrence of the variable's name in the DO loop is replaced by each value of the variable. Here, 'ED_GOTO LINE' currline is replaced by 'ED_GOTO LINE 0', 'ED_GOTO LINE 2', and so on. (Note the position of the quotes in the line 'ED_GOTO LINE' currline).

Note that if the variable's name doesn't appear in the DO loop at all, the variable is useless. For example, if you just wanted the same set of commands to be repeated 64 times, you need only use:

```
DO 64
<insert commands here>
END
```

(You may have noticed that this script only works for blocks 64 lines long. We'll adapt the script to work for any length of block later).

iii) Don't forget to mark the end of the DO loop with END! Also notice that the part between DO and END has been indented (moved to the right) by three spaces. I'd recommend this, it makes for an easier read.

Oh yes, I promised to explain the line 'ED_SETDATA NOTE 27'. It changes the current note to D-3: the number 27 represents the note D-3. This works by simply counting up from 1. 1 = C-1, 2 = C#1, 3 = D-2, .. 12 = B-1, 13 = C-2, 14 = C#2 and so on. Count up far enough and you'll reach 27 = D-3. (By the way, 0 = ---, the empty note).

IF...THEN statements (how to choose how to act depending on a result)

ARexx, being a friendly system, doesn't always demand OctaMED to do things; it can also ask questions. In the midst of a script, you may need to know the current Tracker editor line, or the name of the current instrument, or whether or not the song has been edited since last saving. Indeed, the very first OctaMED command we entered asked the program its version number.

ARexx's IF and THEN devices enable you to choose a course of action depending on OctaMED's answer to a question. We'll look at how to use IF and THEN, and at how to ask the questions in the first place. Our final ARexx script in this brief tour will set all instruments' Hold value to 6, but it will ignore empty instrument slots.

(31) Load the Ed text editor in the usual way. Call this script rexx:sethold.omed. Here it is:

```
/* SetHold.omed: Sets all instruments' hold values to 6 */
/*               (ignoring empty instruments) */

address octamed_rexx
options results

do inst = 1 to 63
   'in_select' inst

   'in_isslotused'
   if result = 1 then 'in_setholddecay hold 6'
end
```

```
'in_select 1'
'wi_showstring Hold values set.'
```

Save and quit when you've finished.

(32) Using the Keyboard Shortcuts window, set Shift-F3 to execute sethold. (Set Raw to 82. Look back at steps 22 to 25 if you're unsure).

(33) Now press Shift-F3. Open the Instrument Parameters window to check that all instruments' Hold values are now 6, apart from empty slots.

So how does this one work? A DO loop is set up, with variable inst running from 1 to 63 in steps of 1 (remember: if BY is left out, the step size is 1). Each instrument is selected in turn by replacing 'IN_SELECT' inst with 'IN_SELECT 1', 'IN_SELECT 2' and so on. (Note that IN_SELECT uses normal decimal values: IN_SELECT 10 selects instrument 0A rather than 10).

Now comes the new part. You can perhaps guess its meaning without further explanation. Is slot used? If result = 1, then set hold to 6. What actually happens is the following:

(a) IN_ISSLOTUSED asks OctaMED if the current instrument slot has been used (i.e. isn't empty). OctaMED answers by giving the variable result a value. It becomes 1 if the slot is used, or 0 if the slot is empty.

(b) The next line is straightforward. IF the variable result is 1, i.e. the slot is used, THEN set the current instrument's Hold to 6. This implies that if the result is 0, i.e. the slot is empty, then don't touch the Hold value and continue with the next instrument instead.

So there are two main points to be made:

i) OctaMED's 'question' commands, such as IN_ISSLOTUSED, return their answer in the variable result.

In fact, they can be returned in any variable you wish, with a little more effort. Remember our second script, Track3D3? I mentioned at the time that the script only worked for 64-line blocks. Well, try making these two alterations:

I) Before DO currline =..., insert: 'ed_getnumlines var numlines'
II) Change TO 63 to TO numlines - 1

The command in step I returns the number of lines in the current block in variable numlines (notice the use of VAR). In step II, currline now runs from 0 to the number of lines in the block minus 1. (Why minus 1? In a 64-line block, the last line number is 63. So...)

By using the same technique, you can get our most recent script to stop when it reaches the last used instrument, instead of carrying right on to instrument 63. See if you can work out how!
(Hint: The commands IN_SELECT LASTUSED and IN_GETNUMBER VAR last will return the last used instrument in variable last)

ii) The IF and THEN devices can be used to act in various ways depending on the result of an OctaMED command. Try replacing the IF...THEN line in our script with the following:

```
         if result = 1 then do
            'in_setholddecay 6'
            'in_setvolume 32'
         end
         else 'in_setname Empty'
```

Now, if result is 1 then Hold will be set to 6 and the default volume will be set to 32. (Notice the use of DO and END: this is necessary if OctaMED should carry out more than one command. It isn't a DO loop!).

Also, if result is NOT 1 (i.e. if it's 0, or if it's anything else for that matter), then the instrument's name will be changed to Empty. So ELSE tells ARexx what to do if the required condition is not met.

Note that you can use symbols other than '=' with IF...THEN. Try < for less than, > for greater than, <= for less than or equal to, >= for greater than or equal to, or ~= for not equal to. Also use & for and and | for or. For example, IF result = 1 & last < 15 THEN... .

Now you have a go at

Writing a Script

main

1.80 Writing a Script

So you've followed all the examples, and you've (at least, sort of ↩) understood how they work. The next obvious step is to start writing your own scripts. But where on earth do you start?

The answer is quite simple: with an idea. You are composing in OctaMED, and suddenly you think, "Wouldn't it be nice if I could write a script to...". Presto, you have an idea for a script. You're a quarter of the way there already.

But the idea's kind of vague. Next you need to form a precise idea. For example, my original idea for the previous example script was,

 (1) IDEA: "How about a script that sets all instruments' Hold values to 6?"

But then I thought, "Hold on a minute, what if the instrument is empty? Changing an empty instrument's Hold value is no good to anyone!" The key words there are "what if". Think of all the possibilities. All the things that could go wrong. So I came up with:

```
 (2) PRECISE IDEA: "How about a script that sets all instruments'
                    Hold values to 6, but ignores empty instruments?"
```

Now it's time to translate informal English into more formal pseudo-code. This is the transitory stage between your idea and the final script.

 (3) PSEUDO CODE:

```
    One    Let variable inst run from the first to the last instrument
    Two    Select the 'inst' instrument number
    Three  Is the current slot used?
    Four   If so, set the current instrument's Hold value to 6
    Five   If not, continue
```

Finally think, "What commands does OctaMED and ARexx provide to do this?" And come up with your AREXX CODE (step 4).

(A quick word about quotes while I've space! If the variable newname contains a string, use the following command to set the current block's name to newname: 'ED_SETBLOCKNAME "'newname'"' . Don't ask why!)

You might find the ARexx command EXIT useful when dealing with things that may go wrong. For example, if the script is to quit if no range is marked:

```
'RN_ISRANGED'
IF result = 0 THEN DO
   'WI_SHOWSTRING Please select range first.'
   EXIT
END
```

Of course, it's quite rare that scripts work first time (unless they're very simple). Errors do tend to creep in. The trouble with running ARexx scripts directly from OctaMED is that no output is given: if there is an error, you have no way of telling what the error is. Here's one way around this:

(1) Select the Workbench screen, and select Project menu -> Execute Command. No, don't start to type ed, instead type newshell (and press Return)

(2) Assuming your script is called newscript.omed, and you've saved it in the REXX: directory, in the AmigaShell window type the following:
copy rexx:newscript.omed ram:newscript.rexx (and press Return)

(3) Now type rx ram:newscript (and press Return)

RX is a program for running ARexx scripts (as long as they have the extension .rexx, which is why you need to copy it to RAM:). Any error messages output by the script will appear in the AmigaShell window.

Well, I hope you've gained sufficient interest to want to find out more!

What now follows is a *long* list of all OctaMED commands and templates.

Some description is given, but for more detailed explanation it's best to read Teijo Kinnunen's own instructions, titled "OctaMED ARexx.doc" which you will find in the Rexx Drawer.

If you'd like more information on ARexx itself, I would try a shareware ARexxGuide, there is one on the Fish disk and Aminet collections.
The comparatively little I know about ARexx has been gleaned from these very guides, so they come with my highest recommendations!
There are also many, many books on the subject.

Please also read the OctaMED ARexx Commands Guide

Keyboard Appendix

OctaMED Soundstudio Features

main

1.81 keyboard_appendix

APPENDIX

Player Commands

MIDI Commands

Player Commands Reference

Keyboard Shortcuts

Hexadecimal Values

1.82 Player Commands

Four points of note:

```
(1) Both the command type and the command level are in HEXADECIMAL. (Please
    read through Appendix D before continuing, just to make sure...)

(2) Please note the following abbreviations:

    Level 1 = 1st command level digit     Level 2 = 2nd command level digit

(3) When entering command types, you can set the first digit to 1 by holding
    down an Alt key. For example, to enter command 19, position the cursor
    over the second command digit and press Alt-9.

(4) In the list that follows, you'll notice that some commands are indicated
    as being Protracker compatible. When loading Protracker modules, OctaMED
    makes any necessary command conversions automatically.
```

Player commands can be split into two groups: normal commands, and MIDI commands, for a list of MIDI commands, see the Manual.guide

Command 00: ARPEGGIO

This command changes the pitch of the note quickly (once every tick). It's usually used to create special effects. If you've ever listened to Commodore 64 music, you must have heard arpeggios!

The pitch is changed between three different pitches (1 to 3) during each note. Level 1 contains the number of halfsteps between pitch 1 and pitch 2, level 2 the number of halfsteps between pitch 1 and pitch 3.

For example, to create a C major arpeggio:

```
   Pitch 1:  C-2
   Pitch 2:  E-2 = 4 halfsteps higher than pitch 1
   Pitch 3:  G-2 = 7 halfsteps higher than pitch 1
```

```
This produces:  C-2 10047
                --- 00047
                --- 00047 and so on... Note that the command level with a
                          blank note (---) continues the arpeggio.
```

As with most commands, experimentation often produces the best results. This effect can sound pretty bad with some instruments but pretty good with others. Use command level 37 for a minor arpeggio, 57 for a sustained 4th.

Commands 01 and 02: SLIDE PITCH UP and DOWN

These two commands slide the pitch of the current track up or down. They actually work by changing the period of the note the amount of the command level every tick. Sounds a bit technical? Dead right! It’s best to leave the donkey work to the automatic slide creator.

Command 03: PORTAMENTO

Again a slide command, and again it’s best to use:
Edit menu -> Pitch Slide -> Type 1.
The difference between this and commands 01 and 02 is that the
latter replay the note they slide to: command 03 doesn’t replay the note.
The command level is the slide speed. Here’s an example:

```
   C-2 50000  <= Note C is played
   E-2 50305  <= This note isn’t played! Instead, the slide target note is
   --- 00300     set to E-2 with a slide speed of 5
   --- 00300  <= Level 00 uses the previous speed (5 here)
```

This example would slide from C-2 to E-2, but the slide stops dead when E-2 is reached (with commands 01 and 02 you can slide beyond the note).

Command 04: VIBRATO

Rapidly slides the pitch up and down (causing a ’wobble’). Level 1 is the vibrato’s speed, level 2 the depth (the length of the pitch slide).

```
   E-3 4043B  <= Low speed, high depth
   --- 004A1  <= High speed, low depth
   --- 00400  <= Level 00 uses the previous speed and/or depth (A1 here)
```

Command 05: SLIDE PITCH AND FADE (Protracker-compatible)

This command combines commands 0300 and 0Dxx. The slide continues at the last speed entered with command 03, and the command level is the fade speed.

```
   C-1 10000
   D-3 10303  <= Slide pitch at speed 3
   --- 00300  <= Continue sliding
   --- 00502  <= Continue sliding + fade at speed 2 (see command 0D later)
```

Command 06: VIBRATO AND FADE (Protracker-compatible)

Combines commands 0400 and 0Dxx. Again, the vibrato continues at the previous speed and depth, and the command level is the fade speed.

Command 07: TREMOLO (Protracker-compatible)

This command is a kind of volume vibrato. Level 1 is the tremolo speed (the higher the value, the greater the speed), and level 2 is the depth. The depth must be quite high before you notice the effect (try A - F).

Command 08: HOLD AND DECAY

Sets the hold and decay values for the note (press Help and click Windows then Instrument Parameters). Level 1 is decay, level 2 is hold.

Command 09: TPL SLIDER

This command sets the TPL slider (the number of ticks per line, The command level must be $01 - $20 ($ means hexadecimal).

Command 0A: VOLUME SLIDE (equivalent to 0D: for Tracker compatibility only)

Please use command 0D instead of this.

Command 0B: PLAYING SEQUENCE POSITION JUMP

This command allows you to create songs that have an introduction, which is only played once, and a main part, which is repeated over and over again. OctaMED jumps to the entry in the playing sequence given by the command level. You must, however, remember that level 00 is the first entry, and 01 the second, so e.g. 03 will be the fourth.

```
   --- 00B06  <= Jump to playing sequence entry 007
```

Note that when more than one playing sequence is defined (i.e. sections), the specified entry in the CURRENT playing sequence is jumped to. At present, no command exists to jump to an entry in the section list.

Command 0C: SET VOLUME

Without doubt the most commonly-used player command. It overrides the

default volume of the instrument.
For example, A-3 40C32 is played with volume 32.

There are 65 volume levels (0 - 64): 0 = silent, 64 = maximum. You may enter the level in either decimal or hex, depending on the state of the cycle gadget in the Song Options window.

It’s also possible to change the volume of an already played note:

```
   A-3 40000
   --- 00000
   --- 00C10  <= Change note A-3’s volume to 10
```

In addition, with this command you may actually set the default instrument volume. The value MUST be in HEX!, and is between 80 and C0: 80 = level $0, C0 = level $40 (hex). In other words, add 80 to the normal volume in hex.

Command 0D: VOLUME SLIDE

You can increase or decrease volume smoothly with this command. Level 1 increases the volume, level 2 decreases. If both level digits have a value greater than zero, level 2 is ignored.

```
   C-2 30C32  <= Sets volume to 32
   --- 00D01  <= Decreases volume slightly
   --- 00D01  <= Decreases volume a little more
   --- 00D20  <= Increases volume back up to 32
```

For techies only: in this command the volume is changed every tick.
So if the TPL tempo slider were 6, a decrease volume value of 1 would lower the volume by 6. Using 0C commands, the above example would look like this (except it doesn’t sound as smooth):

```
   C-2 30C32  (volumes are decimal in this example)
   --- 00C26  <= Subtract 6
   --- 00C20  <= Subtract 6
   --- 00C32  <= Add 12
```

Command 0E: SYNTH JUMP

Only applies to synthetic or hybrid instruments. It is used to trigger a jump in the waveform sequence. The command level is the line number you wish to jump to.

```
   C-4 40000  <= Starts playing the synthetic or hybrid sound
   --- 00000
   --- 00E05  <= Jumps to waveform line number 5
```

Command 0F: PRIMARY TEMPO / MISCELLANEOUS

Command 0F has many different actions depending on the command level.

* Level 00: causes an immediate jump to the next entry in the playing sequence (or to the beginning of the block if you are using Play Block).

It’s actually better to make the block shorter, as it takes less memory, so this function mainly exists for Tracker compatibility.

* Levels 01 - F0: changes the Tempo slider. Levels 01 - 0A really exist for Tracker compatibility, but you should use command 09 instead as it’s directly compatible (but make sure Tempo = 33).

Commands 0FF1 - 0FFF are reserved for special functions, most of which are currently in use:

- 0FF1 makes a single note play twice. Useful for creating fast rhythms. It’s identical to command 1F03. When the TPL slider is 6:

```
C-3 20FF1 is the same as C-3 20000
                         C-3 20000 with double tempo (i.e. 3)
```

- 0FF2 delays the start of a note by half a line (as long as TPL is 6). It’s identical to command 1F30.

- 0FF3 works like 0FF1 except the note is played three times! It’s identical to command 1F02.

- 0FF4 delays the note one-third of a line.

- 0FF5 delays the note two-thirds of a line. Commands 0FF4 and 0FF5 can be used, for example, to create triplets:

```
C-2 10000  Here, the three notes will be evenly spaced throughout
D-2 10FF4  the four lines (they’ll have a length of one-and-a-third
E-2 10FF5  lines each)
--- 00000
```

- 0FF4+5 will only work accurately when TPL is divisable by 3 (3, 6, 9 etc.).

- 0FF8 turns the low-pass filter off.

- 0FF9 turns the low-pass filter on. Commands 0FF8 and 0FF9 work like the Audio Filter Active check box in the Song Options window.

- 0FFA (MIDI only) sends the MIDI hold pedal on command.

- 0FFB (MIDI only) sends the MIDI hold pedal off command.

- 0FFD (not MIDI) causes the pitch of the previous note to be set to the new note, but it’s not replayed.

```
C-1 20000  <= Play note
--- 00000
D#2 20FFD  <= Don’t replay the note, just set the pitch to D#2
```

This is especially useful in playing long samples or samples with a loop set (such as strings and choirs), as the undesirable "click" that you can sometimes hear when playing a new note is eliminated.

- 0FFE stops the song playing. (When used in 5 - 8-channel mode, the song’s notes are also stopped). Can be entered by shift-clicking STOP.

- 0FFF stops the note on the current track. Is almost like 0C00 with Amiga samples, but while 0C00 sets the note's volume to zero, 0FFF turns the note off completely (there is a subtle difference :-).

Commands 11 and 12: SLIDE PITCH UP and DOWN ONCE (Protracker cmds E1 / E2)

Whereas commands 01 and 02 change the pitch on every tick, these commands only change the pitch on the first tick of each line (greater accuracy).

Command 14: PROTRACKER-STYLE VIBRATO

As command 04, except command 14's depth levels produce exactly half the depth as command 04. Protracker pioneered this more accurate command.

Command 15: SET FINETUNE

Overrides the instrument's default finetune value.

Since these are hex numbers, negative numbers must be entered as follows:

```
       -1 => FF        -3 => FD        -5 => FB        -7 => F9
       -2 => FE        -4 => FC        -6 => FA        -8 => F8

   C#3 21503  <= Finetune +3
   D-2 315F8  <= Finetune -8
```

Command 16: REPEAT LINES (LOOP) (Equivalent to Protracker command E6)

This interesting command enables you to set up a repeated section (loop) within a block.

```
-   004 C-3 11600  <= Command level = 00: mark loop beginning
-   005 D#2 10000
-   006 --- 11603  <= Command level = 03: repeat this section three times
```

When the loop has finished, the block continues. You can't nest loops!

Command 18: CUT NOTE (Equivalent to Protracker command E8)

Almost like hold (command 08), except it sets the volume to zero instead of actually switching the note off. (Much the same as the difference between commands 0C00 and 0FFF). The command level must be less than TPL for the command to function.

Command 19: SAMPLE START OFFSET (Protracker command 9)

When playing a sample, this command sets the starting byte in steps of 256 bytes (= $100 in hex). Useful for speech samples.

```
   C-2 11904  <= Play the sample starting at byte $400 = 1024
```

Commands 1A and 1B: SLIDE VOLUME UP / DOWN ONCE (Protracker EA / EB)

Whereas command 0D changes the volume on every tick, these commands only change the volume on the first tick of each line (for greater accuracy).

Command 1D: JUMP TO NEXT PLAYING SEQUENCE ENTRY (Protracker command D)

"But doesn’t command 0F00 jump to the next entry?" Quite right. But with this command you can specify the line number of the first line to be played. The line number is (as usual) given in hex.

```
   --- 01D0A  <= Jump to next playseq entry, and start at line 010 (skip
                 000 - 009)
```

Command 1E: REPLAY LINE (Protracker command EE)

Replays the commands (not the notes) in the current line the specified number of times.

```
   C-2 11B01 --- 01E3F  <= If instrument 01 had a default volume of 64,
                           these two tracks would fade note C-2 away
```

Command 1F: NOTE DELAY AND RETRIGGER (Protracker commands EC and ED)

Gives you accurate control over note playing. You can delay the note any number of ticks, and initiate fast retrigger. Level 1 = note delay value, level 2 = retrigger value.

```
  C-2 11F20  <= Delay 2 ticks  ||  C-2 11F02  <= Retrig every second tick
                               ||
   => tick  0  ---             ||   => tick  0  C-2
            1  ---             ||            1  ---
            2  C-2             ||            2  C-2
            3  ---  etc.       ||            3  ---
                               ||            4  C-2  etc.
```

Command 0FF1 is equivalent to 1F03, 0FF2 to 1F30, and 0FF3 to 1F02.

MIDI COMMANDS

main

1.83 MIDI COMMANDS

This section is a bit like an "erratum" for MIDI, in that it ↩ describes the changes to the above command descriptions required for MIDI use.

Commands 0B, 0F, 16, 1D, 1E, and 1F all work identically with MIDI to Amiga instruments. Commands 15, 18, 19, 1A, and 1B have no effect with MIDI.

If you are unfamiliar with any of the MIDI terms used in this section, please consult your MIDI device's manual.

Commands 05 and 00: CONTROLLER NUMBER and CONTROLLER VALUE

These two commands allow you to change any MIDI controller (command $Bx nn vv, where nn = controller number, vv = value).

Firstly, set the controller number you wish to change with command 05. Then, select the value with command 00. Subsequent uses of command 00 will affect the controller previously set with command 05. Remember that each MIDI channel has its own controller number.

If you need to use a controller value of $00, you can't use 00 as a command level because command 0000 means do nothing. Instead, use level 80. Acceptable controller numbers are $00 - $7F, and values $01 - $80.

Assuming instrument 03 is set to MIDI channel 1:

```
   C-2 30000
   --- 00507  <= Controller $07 = volume (according to the MIDI standard)
   D-2 3007F  <= Sets max. volume ($7F). Command $B0 07 7F is sent (this is
   --- 00000     easier to do with command 17)
   F-3 30001  <= Low volume ($01)
   --- 0055C  <= Sets controller $5C (tremolo depth)...
   G-4 30080  <= ...to 0 (note: level $80 = value $00)
```

Commands 01 and 02: PITCHBENDER UP / DOWN

These commands "turn" the pitchbender 8 x command level steps up or down on every tick. Sounds a bit technical? Indeed, but unfortunately automatic slide making cannot be used. Because the ranges of pitchbenders differ with various MIDI instruments, you must find the right value by experimenting.

The pitchbender isn't automatically reset when a new note is played. To reset it, issue the command again with a level of 00.

```
   C-2 101FF  <= Turns the pitchbender up very quickly...
   --- 00100  <= ...then resets the pitchbender
```

Commands 03 and 13: SET PITCHBENDER

With these commands you may set the pitchbender to an exact value, instead of sliding as with commands 01 and 02. The command level is the pitchbender value, expressed as a signed hexadecimal digit (aaaarrrgh!!! Better read Appendix D again?!?)

```
 $00 = center position,   $80 ($-7F) = smallest value,  $7F = largest value
```

The two commands are identical, except that command 13 can be used when there's a note next to the command: command 03 cannot.

```
   C-2 31380  <= Sets pitchbender to maximum value: use cmd 13 beside notes
   --- 0037F  <= Sets pitchbender to minimum value
   --- 00300  <= Resets pitchbender
```

Command 04: MODULATION WHEEL

Affects the current MIDI channel's modulation wheel. (Modulation is usually vibrato). The command level can be $00 - $7F: $00 = no modulation, $7F = maximum. All notes played using the current MIDI channel will be affected.

Command 08: SET HOLD ONLY

Just to underline that decay can't be used with MIDI instruments. (This is because with MIDI, the volume of an already played note can't be altered).

Command 0A: POLYPHONIC AFTERTOUCH

Changes the polyphonic aftertouch of the most recent note. (Aftertouch is squeezing a key down harder than when it was initially pressed). With this command, each note has an individual amount of pressure (compare with command 0D). The level should be $00 - $7F.

Command 0C: SET VOLUME

Again, just to underline that the volume of a note can't be changed after it has been played (as it can with Amiga samples). The normal Amiga sample volume range of 0 - 64 is converted to the MIDI range of 0 - 127.

Command 0D: CHANNEL AFTERTOUCH

Sends a channel aftertouch message to the most recently used channel. As with command 04, all notes using the most recent channel are given the same pressure (compare with command 0A). The level should be $00 - $7F.

Command 0E: PAN CONTROL

Controls the stereo location of the note. As usual, use levels $00 - $7F.

```
         left        mid        right
          |           |           |
Level:   00          3F          7F
```

Command 10: SEND MIDI MESSAGE

Sends a MIDI message, created using the MIDI message editor. The command level is the message number. Note that with this command the first message is number 0, so you need to subtract one to convert the message number from the message editor for use with this command.

```
   C-2 11003  <= Sends the fourth message, then plays note C-2 (messages are
                 always sent before notes)
```

MIDI timing pulses are sent immediately even if a message is being sent.

NEW Command 17: CHANGE VOLUME CONTROLLER

Changes the volume controller value for the most recently used channel.

```
   --- 00507  <= is the same as =>  --- 0173A
```

--- 0003A

Command 1C: CHANGE MIDI PRESET

Changes the current instrument's MIDI preset number. As with all commands which modify the song parameters, you should be *very* careful when using this command in multi-modules:

That's all the commands OctaMED currently has. The main point, I feel, is that you must experiment with each command until you are happy with each of them. After you've tried them all out, you can push aside the ones you're not so keen on and concentrate on perfecting the ones you do like.

Player Commands Reference

main

1.84 Player commands Reference

Type(s)	Name	Command level (1/2 = level 1/2)
00	Arpeggio	1: 1st pitch, lvl 2: 2nd pitch
01 / 02	Slide pitch up / down	Pitch slide speed
03	Portamento	Pitch slide speed
04	Vibrato	1: speed, 2: depth
05	Slide pitch and fade	Fade speed
06	Vibrato and fade	Fade speed
07	Tremolo	1: speed, 2: depth
08	Hold and decay	1: decay, 2: hold
09	TPL slider	Tempo value
0A	Volume slide	Volume slide speed
0B	Playseq position jump	Position to jump to - 1
0C	Set volume	(Default) volume level
0D	Volume slide	Volume slide speed
0E	Synth jump	Waveform seq. line number
0F	Tempo slider / Miscell.	Various
11 / 12	Slide pitch up / down once	Pitch slide speed
14	Protracker-style vibrato	1: speed, 2: depth
15	Set finetune	Finetune value (signed hex)
16	Loop	Number of times (00 = start)
18	Cut note	Number of ticks
19	Sample start offset	Starting byte $\div$ 256
1A / 1B	Slide volume up / down once	Volume slide speed
1D	Jump to next playseq entry	Starting line number
1E	Replay line	Number of times
1F	Note delay and retrigger	1: delay, 2: retrig.
	MIDI COMMANDS	
05 / 00	Controller number / value	Controller number / value
01 / 02	Pitchbender up / down	Pitchbend speed
03 / 13	Set pitchbender	Pitchbend value (signed hex)

04	Modulation wheel	Modulation level ($00 - $7F)
08	Set hold only	Lvl 1: unused, lvl 2: hold
0A	Polyphonic aftertouch	Aftertouch level ($00 - $7F)
0D	Channel aftertouch	Pressure level ($00 - $7F)
0E	Pan control	Pan value ($00 - $7F)
10	Send MIDI message	Message number - 1
17	Change volume controller	New value
1C	Change MIDI preset	New preset number

Keyboard Shortcuts

main

1.85 Keyboard shorts

IMPORTANT NOTE FOR USERS OF NON-U.S. AND NON-BRITISH KEYBOARDS

The keyboard layout used in this manual corresponds to the standard U.S. keymap (and, by coincidence, to the British keymap). Users of other keyboards, therefore, have slight changes to make to the keys given:

A = Q on a French keyboard (and vice-versa)
Z = W on a French keyboard, and Y on a Swiss or German keyboard (and v-v)

Key(s)	Position is immediately...	Key(s)	Position is immediately...
~	Below Esc	[and]	Right of P
\	Left of Backspace	< and >	Right of M (except French: right of ,)
=	As + (see below)		
/	Left of Right Shift	, and .	Same as < and > above
- and +	Right of 0 (zero)		

* Note that these changes DON'T apply to menu or gadget shortcuts. For example, to open the Input Map Editor window using a French keyboard, hold down the Right Amiga key and press the normal A key (immediately to the right of Tab).

==> Remember that most keyboard shortcuts can be changed using the Keyboard Shortcuts window (see Keyboard Shortcuts in Manual.guide)

Editing shortcuts (used in editing with the Tracker editor)

Esc	Edit mode on/off
Shift-Esc	Chord mode on/off
~	Space mode on/off
Del	Delete note or command digit under cursor
Shift-Del	Delete note and command digits (on current page)
Alt-Del	Delete only command digits (on current page)
Amiga-Del	Delete note and command digits on all pages

Left Amiga-Del	Delete chord. When L-Amiga is held down, every time you press Del the note under the cursor is deleted and the cursor moves to the next selected track. When L-Amiga is released, the cursor advances (as defined in the Keyboard Options window).
Return or A	Insert hold symbol (-|-)
Shift-Return	Insert hold symbols to all tracks of the previous chord
F	Insert / play note at default pitch
Backspace	Delete note and move following notes up
Shift-Backsp	Insert empty note slot
Alt-Backspace	Delete current track
Alt-Shift-Bksp	Insert new track
Amiga-Bksp	Delete line
Sh-Amiga-Bksp	Insert line
Shift- 0 - 9	Enter programmable key 0 - 9
Shift-Ctrl-0-9	Pick note under cursor as programmable key 0 - 9
Tab	Highlight current line
Shift-Tab	Cycle command pages
Ctrl-O	Create volume slide (using command 0C)
Shift-Ctrl-O	Create generic slide (using any command)
Ctrl-T	Create type 1 slide (using command 03)
Shift-Ctrl-T	Create type 2 slide (using commands 01 and 02)
Shift-Alt-Z	Swap block
Shift-Alt-X	Cut block
Shift-Alt-C	Copy block
Shift-Alt-V	Paste block
Ctrl-Z	Erase range
Ctrl-X	Cut range
Ctrl-C	Copy range
Ctrl-V	Paste range
Shift-Ctrl-V	Paste to selected tracks
Ctrl-B	Range current track
Shift-Ctrl-B	Range current block
Ctrl-J	Join block with next
Shift-Ctrl-J	Split block at cursor
Ctrl- <	Swap note under cursor with following note, taking account of the current spacing value
Ctrl- >	Swap notes on adjacent tracks
Ctrl-K	Kill notes to end of track
Shift-Ctrl-K	Kill notes to end of block
Alt-Ctrl-K	Kill notes to end of block and actually remove the deleted part of the block. In other words, the current line becomes the last line of the block.

Cursor movement shortcuts (allow cursor movement in the Tracker editor)

The cursor keys move the cursor one place up/down/left/right.

Alt-<left>	Cursor to previous track	
Alt-<right>	Cursor to next track	
Ctrl-<left>	Previous screenful of tracks	
Ctrl-<right>	Next screenful of tracks	
Shift-Ctrl-<left>	Cursor to track 0	
Shift-Ctrl-<right>	Cursor to last track	
Shift-<up>	Previous block	
Shift-<down>	Next block	
Left Alt-<up>	First block	(Can’t use R.Alt to avoid accidental
Left Alt-<down>	Last block	jump when using Alt-<left>/<right>)
Right Alt-<up>	Up *one* line, whether or not Space is on	
Right Alt-<down>	Down *one* line, whether or not Space is on	
Ctrl-NK(	Previous song	(NK = numeric keypad)
Ctrl-NK)	Next song / Add song	
Shift-Ctrl-NK(	Delete last song	
Shift-Ctrl-NK)	Add song with no confirmation requester	
Alt-Ctrl-NK(	Delete current song	
Alt-Ctrl-NK)	Add and select song (no requester)	
F6	Cursor to first line of block	
F7	Cursor to second quarter of block	
F8	Cursor to middle of block	
F9	Cursor to last quarter of block	
F10	Cursor to last line of block	

Alt-Ctrl-<left>	Go to where sample previously appeared in the song. The sample number is taken from either the number under the cursor, or (if it’s --- 00000) from the current instrument number.
Alt-Ctrl-<right>	Go to where sample next appears in the song

Other shortcuts

Shift-Alt-Space	Play song	Remember that you may also use
Shift-Space	Continue song	the Left Alt shortcuts underlined
Amiga-Space	Play block	in the Main Control window
Alt-Space	Continue block	
Space bar	Stop playing	
Ctrl-Space bar	Reset MIDI presets / pitchbenders / mod. wheels	
Shift-<left>	Previous sample	
Shift-<right>	Next sample	
Alt-Shift-<left>	16 samples backward	
Alt-Shift-<right>	16 samples forward	

Shift-Ctrl- <	Previous free sample slot
Shift-Ctrl- >	Next free sample slot
F1 - F5	Select octaves 1+2 - 5+6 in normal mode, 3+4/2+3/1+2 - 7+8/8+9/9+A in MIDI mode (pressing F1 and F5 cycles through several octaves)
Ctrl- - (minus)	Decrease Tempo slider
Ctrl- + (plus)	Increase Tempo slider
Shift-Ctrl- -	Decrease TPL slider
Shift-Ctrl- +	Increase TPL slider
Ctrl-I	Insert new block
Shift-Ctrl-I	Insert new default block
Ctrl-N	Append new block
Shift-Ctrl-N	Append new default block
Alt-Ctrl-N	Append and select new block
Shift-Alt-Ctrl-N	Append and select new default block
Ctrl-D	Delete current block
Shift-Ctrl-D	Delete last block
Ctrl-S	Save IFF instrument
Shift-Ctrl-S	Save raw instrument
Ctrl-G	Flush current instrument
Shift-Ctrl-G	Flush all unused instruments
Ctrl-A	Automatic Advance Down on/off (Keyboard Options)
Shift-Ctrl-A	Automatic Advance Cursor Right on/off
Alt-Ctrl-A	Automatic Advance to Next Track on/off
Ctrl-W	Advance with sound on/off
Ctrl-F	Display free memory
Alt-~	Set spacing value to length of current range - 1. For example, marking a range from 000 - 002 sets spacing to 2. (It’s logical to subtract 1 since with a spacing of 2, the cursor skips from line 000 to 002 which is 3 lines long).

The following shortcuts apply to the numeric keypad, and all affect the playing sequence (note: playseq = playing sequence).

Shortcut	Effect	Shortcut	Effect
Ctrl-4	Decrease current playseq entry	Ctrl-8	Scroll playseq up
Ctrl-6	Increase current playseq entry	Ctrl-2	Scroll playseq down
Ctrl-5	Insert current block to playseq	Ctrl-7	Top of playseq
Ctrl-0	Duplicate current playseq entry	Ctrl-1	Bottom of playseq
Ctrl-.	Delete current playseq entry		

* In numeric / text boxes:

Shift-<left>	Move to beginning of box (also Ctrl-A)
Shift-<right>	Move to end of box (also Ctrl-F)
Amiga-Q	Restore box’s initial contents

```
    Ctrl-X          Delete box's contents (also Amiga-X)
    Shift-Bksp      Delete from cursor to beginning of box (also Ctrl-U)
    Shift-Del       Delete from cursor to end of box (also Ctrl-K)
    Ctrl-W          Delete current word
```

In addition, the Tab key accepts the edited information (like Return) and activates the next text or numeric box in the window. Shift-Tab accepts and activates the previous box. Remember to press Tab, Shift-Tab or Return after editing a box's contents.

Hexadecimal Values

main

1.86 Hexadecimal Values

Hexadecimal (or 'hex' for short) is basically just a different way ↩ of representing numbers. It is more convenient for the computer because of the way it works, and allows the user to specify a greater range of numbers using the same amount of digits.

Hex values are used in the player commands, MIDI message editor, and the synth editor, so they're quite important in OctaMED.

==> In the usual decimal system, a digit can be ten different values: 0, 1, 2, 3, 4, 5, 6, 7, 8 or 9.

In the hex system, however, a digit may have sixteen values: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E or F. The decimal numbers 10 - 15 are represented by the letters A to F:

```
        Decimal  10 = A;  11 = B;  12 = C;  13 = D;  14 = E;  15 = F
```

With two digits in a number, the decimal system can represent 10 x 10 = 100 different values. The hex system, however, can represent 16 x 16 = 256 values: well over twice the amount of the decimal system. (The lowest hex number is 00 = zero, and the highest is FF = 255 decimal).

Converting between the two systems

So although it feels strange to work in at first, it has its advantages. In a two digit hex number (e.g. 8A), the first digit represents multiples of 16. So, to convert a two-digit hex number to decimal:

```
        Decimal number = (Hex digit 1) x 16 + (hex digit 2)
```

And to convert decimal to hex, divide the decimal number by 16. The quotient is hex digit 1, the remainder is hex digit 2.

e.g. (1) Hex 8A -> decimal: Hex digit 1 = 8, hex digit 2 = A (10).

Decimal number = (8 x 16) + 10 = 106.

(2) Decimal 200 -> hex: 200 $\div$ 16 = 12 remainder 8.
Hex digit 1 = C (12), hex digit 2 = 8.
So hex number = C8.

Hex numbers are sometimes distinguished from decimal numbers by preceding hex with a $ sign. For example: $C8, $FF.

Signed hexadecimal

There is a further complication! Luckily in OctaMED you'll only need to use this in two player commands: MIDI command 03 (set pitchbender) and command 15 (set finetune). (See pages 86 and 83 respectively)

The sign of a number denotes whether the number is positive or negative: that is, the + and - signs.

In hexadecimal, however, there are no - signs. So, negative numbers are represented by positive numbers (it will become clearer!).

==> In signed hex, the numbers $00 - $7F are positive as usual: they represent the decimal numbers 0 - 127. However, the numbers $80 - $FF represent the decimal values -1 to -128:

```
                       Decimal    -1 = $FF
                                  -2 = $FE
                                  -3 = $FD
                                  -4 = $FC etc.

                                 -16 = $F0
                                 -17 = $EF
                                 -18 = $EE etc.

                                -126 = $82
                                -127 = $81
                                -128 = $80
```

So to convert negative decimal numbers to signed hex numbers, first add 256 to the number, then convert to hex as above.

e.g. Decimal -67 -> signed hex: -67 + 256 = 189.
õ189 $\div$ 16 = 11 remainder 13.
Hex digit 1 = 11 (B), hex digit 2 = 13 (C)
So signed hex number = $BC.

Hex numbers will crop up in many areas of computing, so if you've never worked with them before, it's a good idea to get used to them!

Keyboard Appendix

Full ARexx Commands
Soundstudios New Features

main

1.87 Full ARexx Commands

OCTAMED’S AREXX COMMANDS

Option names

Window Names

VE and WI Commands

PL (Playing) Commands

IN (Instrument) commands

SA (Sample editor) commands

SG (Song) Commands

OP (Options) commands

SY (Synth editor) commands

ED (Edit) commands

ED (Edit) Other Commands

RN (Range) commands

1.88 Option names

(use with e.g. OP_GET/SET.

```
AddPaths                  AdvCurs                   AdvLine
AdvTrk                    AdvVoice                  AudioChannels
AuraActive                AuraFixedRate             AuraMinPer
AuraSingleCh              AutoFlush                 AutoFreeze
AutoTerminate             Chord                     CloseWB
Compression               CreateIcons               CreateIconsForSamples
DefHQMode                 DefLPB                    DefSlowHQ
DefTempo                  DefTempoBPM               DefTPL
DefVolMode                DestrSpc                  Edit
ExpShrk_Factor            F6_F10_Highlights         FollowSeq
FourBitPalette            HalveLoadedSamples        HelpViewer
HtoB                      InstCtrlStrLoad           KeymapActive
KeypadMap                 KeypadMode                KillBx7B00
```

```
KillNOff                   LMB                        MIDIActive
MIDIActSensing             MIDIExtSync                MIDIInput
MIDIReadKeyUps             MIDIReadVolume             MIDISendOutInput
MIDISendSync               MMB                        OverwriteReq
Palette_WB                 PolyPlay                   PlayAfterLoading
PPBuffer                   PPEfficiency               RemPaths
ReqTools                   ResetChord                 RMB
RoundSpc                   SaveAllInfo                SaveInstruments
SaveMultiMod               SaveNotation               SaveTimer
SaveTimerOpenWin           ShowUnusedBlocks           SizeOnlyWindowZoom
SMFL_CmdPage_ChAftertouch  SMFL_CmdPage_ChannelVol    SMFL_CmdPage_MIDIPreset
SMFL_CmdPage_ModWheel      SMFL_CmdPage_Pan           SMFL_CmdPage_Pitchbend
SMFL_CmdPage_Tempo         SMFL_CmdPage_Velocity      SMFL_DontIntermix
SMFL_LinesPerBlock         SMFL_MaxTracks             SMFL_OffsetAdjust
SMFL_Resolution            SmpEdCrNoise_Level         SmpEdDensity
SmpEdDigiWVoice            SmpEdEcho_Num              SmpEdEcho_Rate
SmpEdEcho_VolDecr          SmpEdFastGfx               SmpEdFB_Averaging
SmpEdFB_Distance           SmpEdFreehand              SmpEdMaxVolClip
SmpEdMinZoom               SmpEdMix_Dest              SmpEdMix_Source
SmpEdPitch                 SmpEdPitchChg_AntiAlias    SmpEdPitchChg_Dest
SmpEdPitchChg_Source       SmpEdPitchInHz             SmpEdPixelDisp
SmpEdVolChg_End            SmpEdVolChg_Start          SmpEdVolClip
SmpEdZoomRngCent           SmpEd_LoopPoint            Space
SpaceIsDel                 SpaceVal                   SpreadVal
TempoOpChangeCmds          Toccata_LevelDisp          Transpose_InstSlots
WarnDiskFull
```

Window Names

main

1.89 Window Names

(use with WI commands)

```
BLOCKLIST            BLOCKPROPERTIES     EDITOR              EQBAR
EQOSC                EXPANDSHRINK        FONT                HIGHLIGHTOPTIONS
INFORMATION          INSTLOAD            INSTPARAMS          INSTTYPE
INPUTMAP             KEYBOARDOPTIONS     KEYBOARDSHORTCUTS   MAINCONTROL
AURAOPTIONS          MIDIINPUTCHANNEL    MIDIMESSAGEEDITOR   MISCOPTIONS
MOUSEOPTIONS         NEW                 NOTEECHO            PALETTE
PLAYINGSEQ           PROGKEYS            PRINTOPTIONS        SAMPLEEDITOR
SAMPLELIST           SAMPLELISTEDITOR    SAVEOPTIONS         SAVETIMER
SAVE_PPSETTINGS      SAVE_XPKSETTINGS    SA_ADDWORKSPACE     SA_ADJUSTY
SA_CHANGEPITCH       SA_CHANGEVOLUME     SA_CHORDCREATION    SA_NOISE
SA_DISPLAYSETTINGS   SA_ECHO             SA_FILTERBOOST      SA_MIX
SECTIONLIST          SMFLOADOPTIONS      SONGANNOTATION      SONGOPTIONS
SONGSELECT           SPREADNOTES         SYNTHSOUNDEDITOR    SY_PROGRAM
SY_STRETCH           SY_VOLUME           TEMPO               TEMPOOP
TOCCATA_CAPTURE      TRACKVOLUMES        TRANSPOSE
```

VE and WI Commands

main

1.90 VE and WI Commands

```
                (Commands marked * return a value)

*VE_OCTAMED: Returns OctaMED version number
*VE_OCTAMEDREXX: Returns version number of OctaMED ARexx implementation
 WI_ACTIVATE WINDOW/A: Activates the specified window
 WI_CHANGEBOX WINDOW/A,X/N,Y/N,W=WIDTH/N,H=HEIGHT/N: Changes win.
dimensions
 WI_CLOSE WINDOW/A: Closes a window
*WI_ISOPEN WINDOW/A: Tests if the specified window is currently open.
 WI_OCTAMEDTOBACK: Sends the OctaMED screen to back.
 WI_OCTAMEDTOFRONT: Brings the OctaMED screen to front.
 WI_OPEN WINDOW/A: Open a window. See above for a list of supported names.
 WI_OPENSHELL: Opens a new command shell window. Can be more than one open.
*WI_REQUEST BODYTEXT/A,GADGETTEXT/A: Displays a requester with one or more
    gadgets. GADGETTEXT specifies the gadget labels, separated by "|".
    Result: 0 for rightmost choice; 1, 2.. for other choices (left to right)
 WI_SHOWSTRING MESSAGE/F: Displays a message on OctaMED's title bar. If no
    message is specified, resets the title bar to its default state.
 WI_TOBACK WINDOW/A: Moves specified window behind all other windows.
 WI_TOFRONT WINDOW/A: Moves specified window in front of all other windows.
 WI_ZIP WINDOW/A: Shrinks or unshrinks the specified window.
```

PL (Playing) Commands

main

1.91 PL (Playing) Commands

```
                 PL_CONTBLOCK: Play block (continue from current position).

 PL_CONTSONG: Play song (continue from current position).
*PL_GETSTATE: Get player's current state. Returns STOPPED/PLAYSONG/PLAYBLOCK
 PL_JUMPTOTIMERZERO: Jumps to the position where the timer was last reset.
 PL_NEXTALPHAKEYCHANGESINSTR: "Alpha-enter" facility
 PL_PLAYBLOCK: Play block (from the beginning).
 PL_PLAYSONG: Play song (from the beginning).
 PL_PLAYSTATECOMMAND CMD,ELSE: If playing, executes CMD; if not, ELSE.
 PL_RESETMIDI: Resets MIDI pitchbenders, mod. wheels and inst. presets.
 PL_RESETTIMER: Resets the play timer.
 PL_STOP: Stop playing.
```

IN (Instrument) Commands

```
                main
```

1.92 IN (Instrument) commands

```
                IN_FLUSH CURR=CURRENT/S,ALL/S,UN=UNUSED/S: Flush one or more  ↩
                   instruments.
*IN_GETFINETUNE: Returns current instrument’s finetune value (-8 - 7).
*IN_GETHOLD: Returns Hold value.
*IN_GETDECAY: Returns Decay value.
*IN_GETDEFAULTPITCH: Returns default pitch (note number, 0 = none).
*IN_GETEXTENDEDPRESET: Returns 0 if Extended Preset off, 1 if on.
*IN_GETMIDICHANNEL: Returns instrument’s MIDI channel (1 - 16, 0 = none).
*IN_GETMIDIPRESET: Returns MIDI preset.
*IN_GETNAME: Returns current instrument’s name.
*IN_GETNUMBER: Returns number of current instrument (1 - 63).
*IN_GETNUMOCTAVES: Returns number of octaves in the current sample (1 - 7).
   Returns 0 if the instrument is not of type SAMPLE.
*IN_GETOUTPUT: Returns current sample’s output device (AMIGA/AURA/TOCCATA)
*IN_GETSUPPRESSNOTEOFF: Returns 0 if Suppress Note Off is off, 1 if on.
*IN_GETTRANSPOSE: Returns transpose value (-128 - 127).
*IN_GETTYPE: Returns EMPTY/SAMPLE/SYNTH/HYBRID/EXTSAMPLE/SAMPLE16/UNKNOWN.
*IN_GETVOLUME: Instrument’s volume (0 - 64).
*IN_ISSLOTUSED SL=SLOT/N: Returns 1 if the specified instrument slot (1-63)
   is used, 0 if empty. Caution: could be a MIDI instrument even if empty.
*IN_ISSTEREO: Returns 1 if sample in stereo, 0 if mono (or not a sample)
 IN_LOAD NAME/A,NOSL=NOSLIST/S,CN=COMPARENAMES/S: Loads an inst. from disk.
   NOSLIST = don’t search the sample list; COMPARENAMES = don’t load if
   NAME and the name of the current instrument are equal.
 IN_LOADREQ: Opens the Load Instruments requester.
 IN_MONO: Convert this sample to mono.
 IN_SAVE NAME,RAW/S,EIGHTSVX/S,MAUD/S,AIFF/S,WAVE/S: Saves sample. NAME
   should be a full path name. Default format = EIGHTSVX or MAUD.
 IN_SAVEREQ RAW/S,EIGHTSVX/S,MAUD/S,AIFF/S,WAVE/S: Opens save requester for
   saving in the specified format (default: EIGHTSVX).
 IN_SELECT INS/N,NEXT/S,PREV/S,LAST/S,LU=LASTUSED/S,NEXTFREE/S,PREVFREE/S,

            MINUS16/S,PLUS16/S,PREVFOUND=NEARCURSOR/S,SWITCH/S:

   Changes current instrument number. e.g. INS 3 = go to instrument 3;
   NEXTFREE / PREVFREE = next/previous free instrument slot;
   MINUS16 / PLUS16 = go back/forward 16 instruments; PREVFOUND selects
   instrument nearest cursor; SWITCH toggles first inst no. (e.g. 05 -> 15)

 IN_SETDEFAULTPITCH NO=NOTE/N/A: Sets instrument’s default pitch.
 IN_SETFINETUNE FT=FINETUNE/N/A: Sets finetune (-8 - 7).
 IN_SETHOLDDECAY H=HOLD/N,D=DECAY/N: Sets hold and/or decay value(s).
 IN_SETMIDICHANNEL CH=CHANNEL/N/A: Set instrument’s MIDI channel (0 - 16).
 IN_SETMIDIPRESET PR=PRESET/N/A,EXT=EXTENDED/S: Sets MIDI preset.
   EXTENDED = use extended preset.
 IN_SETNAME NAME/A: Sets current instrument’s name (up to 39 characters).
 IN_SETNUMOCTAVES OCTS/N/A: Set number of octaves (1 - 7). Only valid if
   the instrument type is SAMPLE.
 IN_SETOUTPUT AMIGA/S,AURA/S,TOCCATA/S: Sets instrument’s output device.
 IN_SETSUPPRESSNOTEOFF ON/S,OFF/S,TOGGLE/S: Set/clear/toggle Suppr. NoteOff.
 IN_SETTRANSPOSE TR=TRANSPOSE/N/A: Sets transpose value (-128 - 127).
 IN_SETTYPE SAMPLE/S,EXTSAMPLE/S,SYNTHETIC/S,HYBRID/S,SAMPLE16/S: Changes
    instrument type. Not all type changes are allowed (see docs on disk).
 IN_SETVOLUME VOL/N,INCREASE/S,DECREASE/S: Sets instrument’s volume (0-64),
    or optionally INCREASE or DECREASE it by one.
 IN_STEREO: Convert this sample to stereo.
 IN_SWAPINSTRUMENTS SLOT1/N/A,SLOT2/N/A: Swaps the two given instruments.
    It only exchanges the instruments, song data is not modified.


                SA (Sample editor) commands

                main
```

1.93 SA (Sample editor) commands

```
                SA_ADJUSTYRANGE S=SHIFT/N/A: Adjusts range about x-axis (zero line).
    SHIFT is an 8-bit adjust value.
 SA_BOOST AVG=AVERAGING/N,D=DISTANCE/N: Boosts range. If either or both
    values aren’t supplied, the values in the Filter/Boost window are used.
 SA_BUFFERTOSAMPLE: Copies the copy buffer to the sample.
 SA_CENTRALIZESAMPLE: Centralizes entire sample about x-axis.
 SA_CHANGEPITCH S=SOURCE/N,D=DEST/N,AA=ANTIALIAS/S: Changes current sample’s
    pitch. SOURCE and DEST are period values. If not supplied, uses values
    in Change Pitch window. ANTIALIAS turns antialiasing on (otherwise off).
 SA_CHANGEVOL S=START/N,E=END/N,CLIP/S,NOCLIP/S: Changes range volume.
    START and END are percentage values. CLIP/NOCLIP set Change Volume
    window’s Check Clip gadget, otherwise current setting is used. To
    maximize the volume, use SA_CHANGEVOL 999 999 NOCLIP
 SA_CHANGESIZE SIZE/N/A,C=CLEAR/S,RE=RETAINATEND/S,REQ/S: Changes current
    sample’s size. By default, retains existing sample at the start. CLEAR =
    entire sample cleared, RETAINATEND = retain at end, REQ = request user.
 SA_CHOPRANGE: Chops the selected range.
 SA_CLEARRANGE: Clears the selected range.
 SA_COPYRANGE: Copies the selected range to the copy buffer.
 SA_CREATECHORD

N=NOTES/N/M/A,FV=FULLVOL/S,NFV=NOFULLVOL/S,ET=ERASETRAIL/S,

    NET=NOERASETRAIL/S: Creates chord of given notes. NOTES =
    chord’s note numbers (up to 4 can be given). FULLVOL/NOFULLVOL override
    Chord Creation’s Full Volume; ERASETRAIL/NOERASETRAIL override Erase
    Trailing Notes. For example, to create a three-note major chord:
    SA_CREATECHORD 1 5 8

 SA_CREATENOISE L=LEVEL/N: Adds noise to the selected range.
 SA_CUTRANGE: Cuts the selected range.
 SA_DELRANGE: Erases the selected range.
 SA_ECHO NUM/N,R=RATE/N,VD=VOLDECR/N: Creates echoes. Parameters’ purpose
    given in Echo Window instructions (see the on-line help).
 SA_ERASE TS=TOSTART/S,TE=TOEND/S,BL=BEFORELOOP/S,AL=AFTERLOOP/S:
    Erases part of current sample. TOSTART = Range start to sample start;
    TOEND (default) = Range end to sample end; BEFORELOOP = Loop start to
    sample start; AFTERLOOP = Loop end to sample end.
 SA_FILTER AVG=AVERAGING/N,D=DISTANCE/N: Filters range.
*SA_GETBUFFERLENGTH: Returns copy buffer size (0 if empty).
*SA_GETDISPLAYCHANNEL: Returns currently displayed channel of a stereo
    sample (LEFT/RIGHT/BOTH).
*SA_GETDISPLAYSIZE: The size of the currently displayed part of the sample.
*SA_GETDISPLAYSTART: Position of the display start (from the sample start).
*SA_GETLOOPLENGTH: Returns the length of the loop.
*SA_GETLOOPSTART: Returns the beginning of the loop.
*SA_GETLOOPSTATE: Returns 1 if loop is on, 0 if off.
*SA_GETRANGEEND: Returns the range end position.
*SA_GETRANGESTART: Returns the range start position.
*SA_GETSAMPLE O=OFFSET/N/A: Returns the sample value at byte number OFFSET.
    OFFSET range is 0 to (SAMPLELENGTH-1). Sample value always given
    in 16-bit, even for 8-bit samples (-32768 to 32767).
*SA_GETSAMPLELENGTH: Returns the length of the current sample.
 SA_INVERTRANGE: Flips the selected range upside down.
 SA_MIX SL=SOURCELEVEL/N,DL=DESTLEVEL/N: Mixes copy buffer with range.
 SA_MONITOR AS=ALLOWSAMPLING/S: Starts monitoring. Only the user can close
    the monitoring area. ALLOWSAMPLING = like clicking Digitize.
 SA_PASTE OW=OVERWRITE/S: Pastes buffer at Range Start. Will insert unless
    OVERWRITE specified.
 SA_PLAY D=DISPLAY/S,R=RANGE/S,CB=CBUFF/S,LO=LOOP/S,TU=TUNING/S:
    Plays the display / range / loop / tuning tone.
 SA_RANGE S=START/N,E=END/N,A=ALL/S,LO=LOOP/S: Sets range. START / END =
    Range Start / End values. ALL ranges whole sample, LOOP ranges loop.
 SA_RANGETOSYNTHED: Copies the range to the synth editor wave buffer.
 SA_REFRESH: Refreshes the sample display (see SA_SETSAMPLE).
 SA_REMUNUSEDSPACE: Removes unused space from both ends of the sample.
 SA_REVERSERANGE: Reverses the selected range.
 SA_SAMPLETOBUFFER: Copies the whole sample to the copy buffer.
 SA_SETDISPLAYCHANNEL BOTH/S,L=LEFT/S,R=RIGHT/S: Change stereo display mode.
 SA_SETLOOP S=START/N,L=LENGTH/N,KE=KEEPEND/S,R=RANGE/S,D=DISPLAY/S,
            SA=SAMPLE/S: Sets the loop. START/LENGTH = Repeat / RepLen.
    If only START given, length is unchanged unless KEEPEND set (keeps the
    end position intact). RANGE/DISPLAY/SAMPLE loops the range/disp./sample.
 SA_SETLOOPSTATE ON/S,OFF/S: Turns the loop on or off.
 SA_SETSAMPLE O=OFFSET/N/A,V=VALUE/N/A: For editing the actual sample
    waveform. Sets the sample value at byte number OFFSET. OFFSET range is 0
    to (SAMPLELENGTH-1). Sample value should always be given in 16-bit, even
    for 8-bit samples (-32768 to 32767). Call SA_REFRESH to refresh display.
 SA_SHOW S=START/N,L=LENGTH/N,R=RANGE/S,A=ALL/S,LO=LOOP/S:
    Sets the display to the specified part of the sample. START/LENGTH
    specify absolute bytes; RANGE/LOOP/ALL shows the range/loop/sample.
 SA_SWAPBUFFERWITHSAMPLE: Swaps the copy buffer with the sample.
 SA_ZOOM IN/S,OUT/S: Zooms in/out (havling/doubling the display size).


             SG (Song) Commands

             main
```

1.94 SG (Song) Commands

```
                SG_CLEARALLSONGS: Clears all songs (without confirmation).
 SG_CLEARCURRENTSONG: Clears the current song.
 SG_DELETESONG CURRENT=HERE/S,LAST/S: Deletes CURRENT (default) or LAST
    song of a multi-module. Has no effect if only one song exists.
 SG_DISCARDANNOTEXT: Discards the long annotation text (if it exists).
 SG_FORGETMODIFICATIONS: After using this command, OctaMED thinks there have
    been no changes since song was last saved - careful! (See SG_ISMODIFIED)
*SG_GETANNOSTRING: Returns the current (short) annotation string.
*SG_GETCHANNELMODE: Returns the current channel mode (4 - 8).
*SG_GETCURRENTSONGNUMBER: Returns current song number (1 - NUMBEROFSONGS).
*SG_GETFILESIZE MMD2/S,MMD1/S,MMD0/S,MOD/S,SMF0/S,MULTIMOD/S,CURRSONG/S:
    Calculates and returns the unpacked file size of the multi-module
    (default) or the current song in the desired file format.
*SG_GETFILTER: Returns the state of Audio Filter Active (1 or 0).
*SG_GETHQ: Returns the state of High Quality Mode (1 or 0).
*SG_GETMASTERVOL: Returns the master volume (1 - 64).
*SG_GETNAME: Returns the name of the current song (a string).
*SG_GETNUMBEROFSONGS: Returns the total number of songs.
*SG_GETPLAYTRANSPOSE: Returns the play transpose value (-12 - 12).
*SG_GETSLIDEMODE: Returns the slide mode (NO1ST/STD).
*SG_GETTEMPO: Returns the main SPD or BPM tempo (Tempo slider).
*SG_GETTEMPOLPB: Returns the Lines per Beat value (only used in BPM mode).
*SG_GETTEMPOMODE: Returns the current tempo mode (SPD/BPM).
*SG_GETTEMPOTPL: Returns the Ticks Per Line value (a.k.a. secondary tempo).
*SG_GETTRACKVOL TRACK/N/A: Returns the track volume (1 - 64) of the
    specified track (0 - 63).
*SG_GETVOLMODE: Returns the current volume mode (HEX/DEC).
 SG_GOTOSONG SONGNUM/N,NEXT/S,PREV/S,LAST/S,NEXTORAPPEND/S: Selects a song.
    SONGNUM = song number; NEXT/PREV = next/previous song; LAST = last song.
    NEXTORAPPEND = NEXT with requester to add new song if current is last.
*SG_ISMODIFIED: Returns a non-zero value if the song has been modified since
    it was last loaded or saved. Otherwise returns 0.
*SG_ISTRACKON TRK=TRACK/N: Returns the on/off (1/0) state of the specified
    track. If track not specified, current track is assumed.
*SG_ISTRACKSELECTED TRK=TRACK/N: Returns the selected/deselected (1/0) state
    of the specified track. If track unspecified, current track is assumed.
 SG_LOAD NAME,FORCE/S: Loads song into memory. IF NAME omitted, opens file
    requester. Returns an error if song modified, unless FORCE is set.
 SG_LOADANNOTEXT NAME: Loads annotation text file NAME (or opens requester).
 SG_NEWSONG HERE/S,LAST/S: Adds new song at curr. pos. (HERE) or after LAST.
 SG_SAVE NAME,FULLNAME/K,MMD2/S,MMD1/S,MMD0/S,MOD/S,SMF0/S,MULTIMOD/S,
         CURRSONG/S: Saves current song (CURRSONG) or multi-module
    (MULTIMOD, default) in the given format (MMD2/MMD1/MMD0/MOD/SMF0). Saves
    under name NAME (if given) using current save path (see SG_SETSAVEPATH).
    FULLNAME = save path + name.
 SG_SAVEANNOTEXT NAME: Saves annotation text to NAME (or opens requester).
 SG_SETANNOSTRING STRING/F: Sets/clears (if no STRING) short anno string.
 SG_SETCHANNELMODE MODE/N/A: Sets the current channel mode (4 - 8).
 SG_SETFILTER ON/S,OFF/S: Sets the state of Audio Filter Active (1 or 0).
 SG_SETHQ ON/S,OFF/S: Sets the state of High Quality Mode (1 or 0).
 SG_SETMASTERVOL VOL/N/A: Sets the master volume (1 - 64).
 SG_SETNAME NAME/A/F: Sets the name of the current song.
 SG_SETPLAYTRANSPOSE TRANSP/N/A: Sets the play transpose (-12 - 12).
 SG_SETSAVENAME NAME/A: Sets the current project's default save filename.
```

```
   But use the next command to set the save path.
 SG_SETSAVEPATH PATH/A: Sets the default save file name and save path.
   You should specify the volume name, the directory and the filename.
 SG_SETSLIDEMODE STD/S,NO1ST/S: Sets the slide mode.
 SG_SETTEMPO BPM=SPD/N,TPL/N,LPB/N: Sets the current tempo.
 SG_SETTEMPOMODE SPD/S,BPM/S: Sets the tempo mode (BPM is now recommended).
 SG_SETTRACKSTATE TRK=TRACK/N/A,ON/S,OFF/S,TOGGLE/S,SEL/S,DESEL/S,
                  TOGGLESEL/S: Sets the state of the specified track.
   ON/OFF/TOGGLE = track playing status; other switches = selected state.
 SG_SETTRACKSTATES ON/S,OFF/S: Sets all tracks ON or OFF.
 SG_SETTRACKVOL TRACK/N/A,VOL/N/A: Sets TRACK's (0-63) track volume (1-64).
 SG_SETVOLMODE DEC/S,HEX/S: Sets the volume mode (DECimal or HEX).
 SG_SHOWFREEMEMORY: Displays free memory on the title bar (used for Ctrl-F).
 SG_TEMPO_DECREASE TPL/S: Decreases the Tempo (or TPL if set) slider.
 SG_TEMPO_INCREASE TPL/S: Increases the Tempo (or TPL if set) slider.
 SG_VOLCONVERT TOHEX/S,TODEC/S,REQ/S: Converts the volume commands of the
   song. TOHEX = dec -> hex; TODEC = hex -> dec; REQ asks the user.
```

OP (Options) commands

main

1.95 OP (Options) commands

```
               *OP_GET OPT/A: Returns specified option's current value. (See p ↩
                  .65) See docs
   on disk for returned value details.
*OP_GETKEYBOARDOCT: Returns current octaves: 0 = 1+2, 1 = 2+3, 2 = 3+4 etc.
 OP_LOADKEYBOARDSHORTCUTS NAME/A: Loads shortcut file (NAME = full path).
 OP_LOADSETTINGS NAME/A: Loads settings file (NAME = full path name).
 OP_MULTICMD CMD/M: Multiple commands (see §12.9.6 part 5). You can't use
   commands which need spaces in their parameters (even with quotes).
 OP_REPEATCMD TIMES/N,CMD/F: Repeats command CMD the given number of times.
   Can use with OP_MULTICMD to repeat a sequence of commands.
 OP_SAVEKEYBOARDSHORTCUTS NAME/A: Saves shortcut file (NAME = full path).
 OP_SET OPT/A,VAL/N,ON/S,OFF/S,TOGGLE/S: Changes specified option's value or
   state. ON/OFF/TOGGLE affect check boxes and 'toggling' menu items; VAL
   affects other options. Make sure the value you give is valid!!
 OP_SETKEYBOARDOCT OCT/N,FKEY/K/N: Sets current edit octaves. OCT = 1 - 9
   (1 = 1+2, 2 = 2+3 etc.) or use FKEY 1 - 5: corresponds to key F1 - F5.
 OP_TOGGLEBETWEEN OPT/A,VAL1/N/A,VAL2/N/A: Toggles options between two
   possible values (not check boxes / toggling menu items).
 OP_UPDATE ON/S,OFF/S: Turns Editor and Information updating on/off.
```

SY (Synth editor) commands

main

1.96 SY (Synth editor) commands

```
                SY_ADDWAVES: Adds the right waveform to the left one.
 SY_CHANGEVOL VAL/N/A: Changes range volume (VAL = percentage of original).
 SY_CLEARRANGE: Clears range.
 SY_CLEARWAVE: Clears current waveform.
 SY_COPYRANGE: Copies range to the right waveform.
 SY_COPYWAVE TOTEMP/S,FROMTEMP/S: Copies waveform (TOTEMP = right to left).
 SY_CUTRANGE: Cuts the selected range.
 SY_DELPROGCOMMAND VOL/S,WF/S: Deletes entry from VOLume or WaveForm seq.
 SY_DELWAVE L=LAST/S: Deletes current waveform (unless LAST specified).
 SY_DOUBLERANGE: Doubles range.
*SY_GETNUMBEROFWAVES: Returns number of waveforms in current synthsound.
*SY_GETPROGCOMMAND O=OFFSET/N/A,VOL/S,WF/S: Returns an sequence entry.
    OFFSET specifies the position in either the VOLume or WaveForm sequence.
*SY_GETPROGCURSORX: Returns the horizontal cursor position (0 - 5).
*SY_GETPROGLENGTH VOL/S,WF/S: Returns length of VOL or WF sequence.
*SY_GETPROGLINE: Returns current line number. (May exceed program length).
*SY_GETPROGSPEED VOL/S,WF/S: Returns program (= sequence) execution speed.
*SY_GETSAMPLE O=OFFSET/N/A: Returns a sample value from the current waveform
    at position OFFSET (-32768 to 32767: in 16-bit even though 8-bit!).
*SY_GETSELECTEDWAVE: Returns selected wave (0 = left, 1 = right).
*SY_GETWAVELENGTH: Returns length of curr. waveform (2 - 128, even number).
*SY_GETWAVENUM: Returns the number of the currently selected wave.
 SY_GOTOPROGLINE L=LINE/N/A: Moves to specified sequence line.
 SY_GOTOWAVE WAVE/N,NEXT/S,PREV/S: Moves to specified waveform.
 SY_MIXWAVES: Mixes the right waveform to the left.
 SY_MOVEPROGCURSOR POS/N,LEFT/S,RIGHT/S,UP/S,DOWN/S,FKEY/K/N: Moves sequence
    cursor. POS = horizontal position 1 - 5; FKEY: 0 = F6, 1 = F7 etc.
 SY_NEW: Creates a new synthsound.
 SY_NEWWAVE H=HERE/S: Creates new waveform (adds to end unless HERE given).
 SY_NOISE: Sets the current waveform to the Noise preset.
 SY_PASTERANGE: Pastes the right waveform to the left.
 SY_PULSE: Sets the current waveform to the Pulse preset.
 SY_RAMPDOWN: Sets the current waveform to the Ramp Down preset.
 SY_RAMPUP: Sets the current waveform to the Ramp Up.
 SY_RANGE S=START/N,E=END/N,A=ALL/S: Sets the range.
 SY_REVERSERANGE: Reverses the selected range.
 SY_SETPROGRAM PROGRAM/M/A,VOL/S,WF/S,INS=INSERT/S: Enters a VOL or WF seq.
    Entire sequence is set unless INSERT given. (See docs on disk for more)
 SY_SETPROGSPEED VOL/K/N,WF/K/N: Sets sequence(s) execution speed.
 SY_SETSAMPLE O=OFFSET/N/A,V=VALUE/N/A: Set sample value at position OFFSET
    (-32768 to 32767: 16-bit values). Display refreshed automatically.
 SY_SETSELECTEDWAVE TEMP/S,ACTUAL/S: Selects right (TEMP) or left waveform.
 SY_SETWAVELENGTH LENGTH/N/A: Sets current waveform length (2 - 128).
 SY_SHIFTRANGE L=LEFT/S,R=RIGHT/S: Shifts range left or right.
 SY_SINE: Sets the current waveform to the Sine preset.
 SY_STRETCH DIST/N/A: Stretches curr. waveform at cursor (DIST = distance).
 SY_SWAPWAVES: Exchanges the left and right waveforms.
 SY_TRIANGLE: Sets the current waveform to the Triangle preset.
 SY_UNDO: Undoes (reverses the effects of) the last editing operation.
 SY_WAVETRANSFORMATION START/N/A,END/N/A: Transforms START to END waveform.


                ED (Edit) commands
```

main

1.97 ED (Edit) commands

```
                 ( Playing Sequence and Section List )

 ED_CLEARSEQ FORCE/S: Clears the current playseq. FORCE overrides requester.
 ED_DELETEPLAYSEQ: Deletes the current playseq entry.
 ED_DELETESECLIST: Deletes the current section list entry.
 ED_DELETESECTION: Deletes the current section (playing sequence list).
*ED_GETCURRPLAYSEQ: Returns current playseq position in current section.
*ED_GETCURRSECLIST: Returns the current section list position.
*ED_GETCURRSECTION: Returns the current section number.
*ED_GETNUMPLAYSEQ: Returns the current section's length.
*ED_GETNUMSECLIST: Returns the section list's length.
*ED_GETNUMSECTIONS: Returns the total number of sections.
*ED_GETPLAYSEQBLOCK O=OFFSET/N: Returns the playseq block no. at pos OFFSET.
*ED_GETSECLISTSECTION O=OFFSET/N: Returns seclist section no. at pos OFFSET.
*ED_GETSECTIONNAME: Returns the current section's name.
 ED_GOTOPLAYSEQ NUM/N,PREV/S,NEXT/S,FIRST/S,LAST/S,BOTTOM/S,JUMPED/S,
    JUMPPLAY/S: Changes current playseq position. NUM = absolute position;
    BOTTOM = under the last entry; JUMPED instructs Tracker editor to jump
    to first line of block in that position; JUMPPLAY = same while playing.
 ED_GOTOSECLIST SECPOS/N/A: Changes current section list position.
 ED_GOTOSECTION SECT/N/A: Changes current section.
*ED_ISBLOCKINSEQ B=BLOCK/N: Returns 0 if block BLOCK doesn't exist in any
    section, 1 if it does. (cf. Block List window -> Show Unused)
 ED_NEWPLAYSEQ BLOCK/N,CURRBLOCK/S,COPYBLOCK/S: Inserts new playseq entry
    at current position. CURRBLOCK inserts current block number, COPYBLOCK
    copies current playseq entry.
 ED_NEWSECLIST INSERT=HERE/S,APPEND=LAST/S: Adds new entry to section list.
 ED_NEWSECTION INSERT=HERE/S,APPEND=LAST/S: Creates new section.
 ED_SETPLAYSEQBLOCK O=OFFSET/N,B=BLOCK/N,INCREASE/S,DECREASE/S: Edits the
    current section. INCREASE/DECREASE change selected/current value by 1.
 ED_SETSECLISTSECTION O=OFFSET/N,SC=SECNUM/N: Edits the section list.
 ED_SETSECTIONNAME NAME: Sets the current section's name.
```

ED (Edit) Other Commands

main

1.98 ED (Edit) Other Commands

```
                 ED_ADVANCELINE LINES/N,UP/S,DOWN/S,FKEY/K/N,SPC=SPACING/S,USESETT ↩
                    /S:
    Advances the cursor UP or DOWN lines LINES (default 1). SPACING uses
    current Space Value (ensure Space is on). FKEY (6 - 10) moves to F6 -
    F10 position. USESETT uses current Keyboard Options advance settings.
 ED_CLEARNOTECMD NOTE/S,CMD/S,ALLPAGES/S: Clears note and/or command digits
```

```
    on the current page (default) or ALLPAGES. Takes Space into account.
 ED_CREATESLIDE VOL/S,GEN/S,PITCH1/S,PITCH2/S: Creates slide (Edit menu).
 ED_DELETEBLOCK HERE=CURRENT/S,LAST/S: Deletes CURRENT (def.) or LAST block.
 ED_DELETENOTE B=BLOCK/K/N,L=LINE/K/N,T=TRACK/K/N,DELSPACE/N: Like pressing
    Backspace. Default: current cursor pos. DELSPACE = no. of removed notes.
 ED_ENTERNOTE NOTE/N,PROGKEY/K/N,HOLD/S: Enters NOTE/PROGKEY/HOLD symbol.
 ED_FINDSAMPLEOCCURRENCE PREV/S,NEXT/S: Like Alt-Ctrl-<left>/<right>.
⋆ED_GETBLOCKNAME B=BLOCK/N: Returns current/specified block’s name.
⋆ED_GETCURRBLOCK: Returns the current block number.
⋆ED_GETCURRLINE: Returns the current line number (in decimal).
⋆ED_GETCURRPAGE: Returns the currently displayed command page.
⋆ED_GETCURRTRACK: Returns the track that the cursor is on.
⋆ED_GETDATA B=BLOCK/K/N,L=LINE/K/N,T=TRACK/K/N,P=PAGE/K/N,NOTE/S,INUM/S,
    CMDNUM=CMDTYPE/S,QUAL=CMDLVL/S,PRTALL/S: Returns song data. Default is
    cursor pos but can use BLOCK/LINE/TRACK/PAGE. The switches set what data
    to return (note/inst no/cmd type/cmd level/string e.g. ’C-3 20C48’).
⋆ED_GETLINEHIGHLIGHT B=BLOCK/K/N,L=LINE/N: Returns 1 if line highlighted.
⋆ED_GETNUMBLOCKS: Returns the total number of blocks in memory.
⋆ED_GETNUMLINES B=BLOCK/N: Returns the current/specified block’s length.
⋆ED_GETNUMPAGES B=BLOCK/N: Returns the curr/spec block’s number of pages.
⋆ED_GETNUMTRACKS B=BLOCK/N: Returns the curr/spec block’s number of tracks.
⋆ED_GETPREVIOUSINUM: Returns the current track’s previous instrument number.
 ED_GOTO B=BLOCK/K/N,L=LINE/K/N,T=TRACK/K/N,P=PAGE/K/N: Moves the cursor.
 ED_GOTOBLOCK PREV/S,NEXT/S,FIRST/S,LAST/S,FIRSTLINE/S,FIRSTPAGE/S: Mainly
    for keyboard shortcuts. FIRSTLINE/PAGE jumps to block’s first line/page.
 ED_GOTOCMDPAGE PAGE/N,NEXT/S,PREV/S,CYCLEFWD/S: Selects current command
    page. CYCLEFWD cycles forward, jumping to page 1 after the last page.
 ED_GOTOTRACK PREVNOTE/S,NEXTNOTE/S,NEXTCHORD/S,FIRST/S,LAST/S,PREVDISP/S,
    NEXTDISP/S: Mainly for keyboard shortcuts. NEXTCHORD = next chord note.
 ED_HIGHLIGHTLINE B=BLOCK/K/N,L=LINE/N,ON/S,OFF/S,TOGGLE/S: Highlights,
    dehighlights, or toggles the current or specified line
 ED_INSERTEMPTYNOTE B=BLOCK/K/N,L=LINE/K/N,T=TRACK/K/N,INSSPACE/N: Like
    Shift-Backspace. INSSPACE = number of inserted notes (default = 1).
 ED_KILLNOTES CURRTRACK/S,ALLTRACKS/S,CBS=CHANGEBLOCKSIZE/S: Kills notes
    below cursor. (See keyboard shortcuts (Shift/Alt)-Ctrl-K)
 ED_MOVECURSOR
 LEFT/S,RIGHT/S,NOTE/S,INUM/S,CMDNUM1=CMDTYPE1/S,CMDNUM2=
               CMDTYPE2/S,QUAL1=CMDLVL1/S,QUAL2=CMDLVL2/S: Moves cursor.
 ED_NEWBLOCK HERE=INSERT/S,LAST=APPEND/S,TRACKS/N/K,LINES/N/K,PAGES/N/K:
    INSERTs/APPENDs new block. You can also specify its size.
⋆ED_NUMBERTONOTE NOTE/N/A: Converts note number to string. E.g. 2 -> ’C#1’.
 ED_PICKPROGKEYNOTE PROGKEY/N/A: Like keyboard shortcuts Shift-Ctrl- 1 - 0.
 ED_PLAYNOTE NOTE/N,INUM/N,TRACK/N: Plays a note. Defaults = current.
 ED_SETBLOCKLINES B=BLOCK/N/K,LINES/N/A: Sets curr/spec block’s length.
 ED_SETBLOCKNAME B=BLOCK/K/N,NAME: Sets current/specified block’s name.
 ED_SETBLOCKPAGES B=BLOCK/N/K,PAGES/N/A: Sets block’s no. of pages to PAGES.
 ED_SETBLOCKTRACKS B=BLOCK/N/K,TRACKS/N/A: Sets curr/spec block’s tracks.
 ED_SETDATA
B=BLOCK/K/N,L=LINE/K/N,T=TRACK/K/N,P=PAGE/K/N,NOTE/K/N,INUM/K/N,
    CMDNUM=CMDTYPE/K/N,QUAL=CMDLVL/K/N,ALTCMD/S: Changes notes/data (see
    ED_GETDATA). ALTCMD enters the Right Alt programmable key definition.
 ED_SETDATA_UPDATE ON/S,OFF/S: Turns ED_SETDATA updating on/off.
 ED_SWAPNOTEWITH FOLL=FOLLOWING/S,RIGHT/S: Like Ctrl-< / >.
```

RN (Range) commands

main

1.99 RN (Range) commands

```
                 RN_CANCELRANGE: Discards the range.
 RN_CHANGEINUM
SRC/N,DEST/N,CHANGE/S,EXCHANGE=SWAP/S,DELNOTES/S,AFFECT/K/N:
    Changes instrument numbers (see the Transpose window). AFFECT is 1 - 5.
 RN_CHANGENOTES SRC/N,DEST/N,CHANGE/S,EXCHANGE=SWAP/S,AFFECT/K/N: Chg
notes.
 RN_COPY RANGE/S,TRACK/S,BLOCK/S: Copies a RANGE/TRACK/BLOCK.
 RN_CUT RANGE/S,TRACK/S,BLOCK/S: Cuts a RANGE/TRACK/BLOCK.
 RN_DELETELINE LINE/N: Deletes the current or specified line.
 RN_DELETETRACK TRACK/N,WHOLESONG/S: Deletes the current or specified track.
    WHOLESONG deletes the same-numbered track throughout the entire song.
 RN_DISCARDBUFFERS: Discards all copy buffers.
 RN_ERASE RANGE/S,TRACK/S,BLOCK/S: Clears a RANGE/TRACK/BLOCK.
 RN_EXPANDBLOCK FACTOR/N: Expands the block. Can specify FACTOR.
*RN_GETRANGEENDLINE: Returns the line number of the last ranged line.
*RN_GETRANGEENDTRACK: Returns the track number of the last ranged track.
*RN_GETRANGESTARTLINE: Returns the line number of the first ranged line.
*RN_GETRANGESTARTTRACK: Returns the track number of the first ranged track.
 RN_GETSPACINGFROMRANGE: Like keyboard shortcut Alt-~.
 RN_INSERTLINE LINE/N: Inserts an empty line at current/specified line.
 RN_INSERTTRACK TRACK/N,WHOLESONG/S: Inserts an empty track.
*RN_ISRANGED: Returns 1 if a range is marked, 0 if not.
 RN_JOINBLOCKS: Joins the current block with the following one.
 RN_NOTEECHO DIST/N,MINVOL/N: Echoes the range (see the Note Echo window).
 RN_PASTE RANGE/S,TRACK/S,BLOCK/S,TS=TOSELECTED/S: Pastes. (See Edit menu)
 RN_SETRANGE STARTTRACK/N,STARTLINE/N,ENDTRACK/N,ENDLINE/N,TRACK/K/N,
             LINE/K/N,CURRBLOCK/S,CURRTRACK/S: Sets range. Can specify
    absolute values, or range a given line/track, or range current trk/blk.
 RN_SHRINKBLOCK FACTOR/N: Shrinks the block. Can specify FACTOR.
 RN_SPLITBLOCK LINE/N: Splits the block at the current/specified line.
 RN_SPREADNOTES WIDTH/N: Spreads the ranged notes (see Spread Notes window).
 RN_SWAP TRACK/S,BLOCK/S: Swaps a track or block with the copy buffer.
 RN_TRANSPOSE HSUP/S,HSDOWN/S,OCTUP/S,OCTDOWN/S,AFFECT/K/N,ALLINSTRS/S,
    CURRINSTR/S: Transposes (see the Transpose window).


 MM_MOVECURSOR LEFT/S,RIGHT/S,UP/S,DOWN/S: For the MIDI Message Editor.


Well, that’s all of them, have fun!
```

main