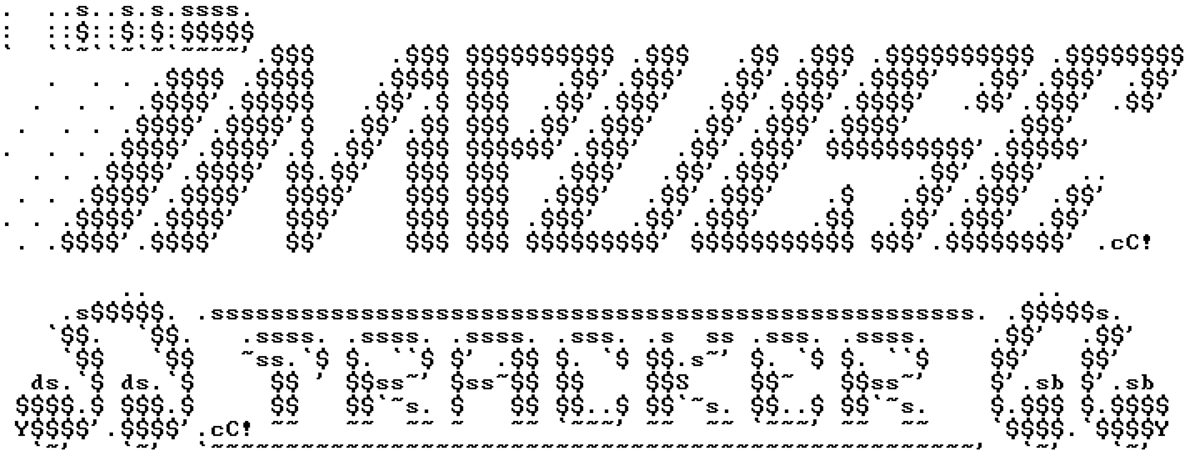
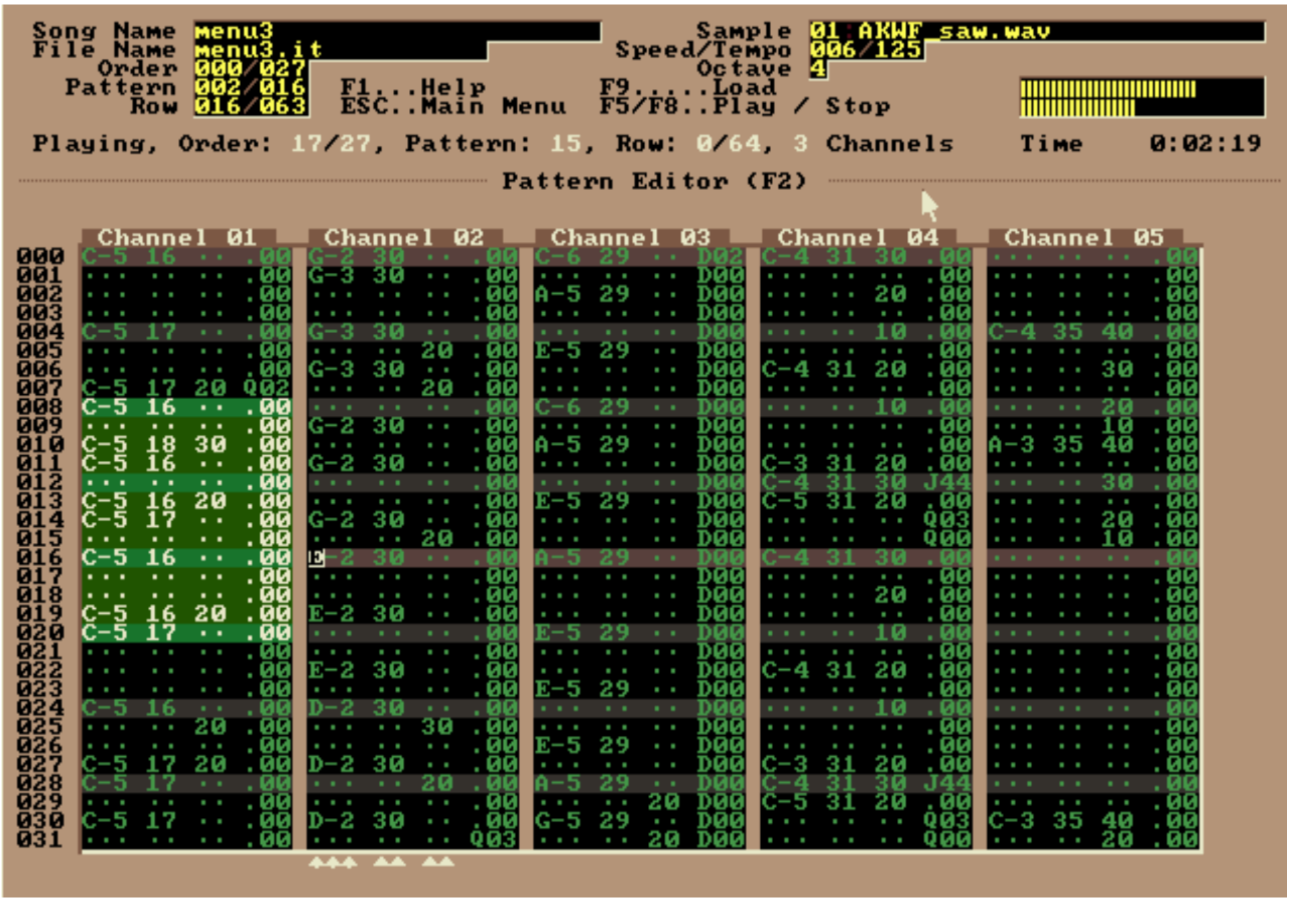


User's Manual



```
  Song Name menu3                                 Sample 01 AKWF_saw.wav
  File Name menu3.it                         Speed/Tempo 006/125
      Order 000/027                               Octave 4
    Pattern 002/016   F1...Help        F9.....Load
        Row 016/063   ESC..Main Menu   F5/F8..Play / Stop

  Playing, Order: 17/27, Pattern: 15, Row: 0/64, 3 Channels     Time    0:02:19
  ............................. Pattern Editor (F2) ............................

      Channel 01      Channel 02      Channel 03      Channel 04      Channel 05
000 C-5 16 ·· .00  G-2 30 ·· .00  C-6 29 ·· D02  C-4 31 30 .00  ··· ·· ·· .00
001 ··· ·· ·· .00  G-3 30 ·· .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
002 ··· ·· ·· .00  ··· ·· ·· .00  A-5 29 ·· D00  ··· ·· 20 .00  ··· ·· ·· .00
003 ··· ·· ·· .00  ··· ·· ·· .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
004 C-5 17 ·· .00  G-3 30 ·· .00  ··· ·· ·· D00  ··· ·· 10 .00  C-4 35 40 .00
005 ··· ·· ·· .00  ··· ·· 20 .00  E-5 29 ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
006 ··· ·· ·· .00  G-3 30 ·· .00  ··· ·· ·· D00  C-4 31 20 .00  ··· ·· 30 .00
007 C-5 17 20 Q02  ··· ·· 20 .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
008 C-5 16 ·· .00  ··· ·· ·· .00  C-6 29 ·· D00  ··· ·· 10 .00  ··· ·· 20 .00
009 ··· ·· ·· .00  G-2 30 ·· .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· 10 .00
010 C-5 18 30 .00  ··· ·· ·· .00  A-5 29 ·· D00  ··· ·· ·· .00  A-3 35 40 .00
011 C-5 16 ·· .00  G-2 30 ·· .00  ··· ·· ·· D00  C-3 31 20 .00  ··· ·· ·· .00
012 ··· ·· ·· .00  ··· ·· ·· .00  ··· ·· ·· D00  C-4 31 30 J44  ··· ·· 30 .00
013 C-5 16 20 .00  ··· ·· ·· .00  E-5 29 ·· D00  C-5 31 20 .00  ··· ·· ·· .00
014 C-5 17 ·· .00  G-2 30 ·· .00  ··· ·· ·· D00  ··· ·· ·· Q03  ··· ·· 20 .00
015 ··· ·· ·· .00  ··· ·· 20 .00  ··· ·· ·· D00  ··· ·· ·· Q00  ··· ·· 10 .00
016 C-5 16 ·· .00  E-2 30 ·· .00  A-5 29 ·· D00  C-4 31 30 .00  ··· ·· ·· .00
017 ··· ·· ·· .00  ··· ·· ·· .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
018 ··· ·· ·· .00  ··· ·· ·· .00  ··· ·· ·· D00  ··· ·· 20 .00  ··· ·· ·· .00
019 C-5 16 20 .00  E-2 30 ·· .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
020 C-5 17 ·· .00  ··· ·· ·· .00  E-5 29 ·· D00  ··· ·· 10 .00  ··· ·· ·· .00
021 ··· ·· ·· .00  ··· ·· ·· .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
022 ··· ·· ·· .00  E-2 30 ·· .00  ··· ·· ·· D00  C-4 31 20 .00  ··· ·· ·· .00
023 ··· ·· ·· .00  ··· ·· ·· .00  E-5 29 ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
024 C-5 16 ·· .00  D-2 30 ·· .00  ··· ·· ·· D00  ··· ·· 10 .00  ··· ·· ·· .00
025 ··· ·· 20 .00  ··· ·· 30 .00  ··· ·· ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
026 ··· ·· ·· .00  ··· ·· ·· .00  E-5 29 ·· D00  ··· ·· ·· .00  ··· ·· ·· .00
027 C-5 17 20 .00  D-2 30 ·· .00  ··· ·· ·· D00  C-3 31 20 .00  ··· ·· ·· .00
028 C-5 17 ·· .00  ··· ·· 20 .00  A-5 29 ·· D00  C-4 31 30 J44  ··· ·· ·· .00
029 ··· ·· ·· .00  ··· ·· ·· .00  ··· ·· 20 D00  C-5 31 20 .00  ··· ·· ·· .00
030 C-5 17 ·· .00  D-2 30 ·· .00  G-5 29 ·· D00  ··· ·· ·· Q03  C-3 35 40 .00
031 ··· ·· ·· .00  ··· ·· ·· Q03  ··· ·· 20 D00  ··· ·· ·· Q00  ··· ·· 20 .00
```

```
.  ..s..s.s.ssss.
:  ::$::$:$:$$$$$
`  ``~``~`~`~~~~'.$$$     .$$$ $$$$$$$$$$ .$$$   .$$ .$$$ .$$$$$$$$$$ .$$$$$$$$
   .  . . .$$$$ .$$$$    .$$$$ $$$   .$$'.$$$'  .$$'.$$$'.$$$$'  .$$'.$$$' .$$'
  .  . . .$$$$'.$$$$$   .$$'.$ $$$  .$$'.$$$'  .$$'.$$$'.$$$$'  .$$'.$$$' .$$'
 .  . . .$$$$'.$$$$'$  .$$'.$$ $$$ .$$'.$$$'  .$$'.$$$'.$$$$'      .$$$'
.  . . .$$$$'.$$$$'.$ .$$' $$$ $$$$$$'.$$$'  .$$'.$$$' $$$$$$$$$$'.$$$$$'
  . . .$$$$'.$$$$' $$.$$'  $$$ $$$   .$$$'  .$$'.$$$'        .$$'.$$$'  ..
 . . .$$$$'.$$$$'  $$$$'   $$$ $$$  .$$$'  .$$'.$$$'   .$   .$$'.$$$' .$$'
. . .$$$$'.$$$$'   $$$'    $$$ $$$ .$$$'  .$$'.$$$'   .$$  .$$'.$$$' .$$'
 . .$$$$'.$$$$'    $$'     $$$ $$$ $$$$$$$$$' $$$$$$$$$$$ $$$'.$$$$$$$$' .cC!

        ..                                                           ..
    .s$$$$$. .sssssssssssssssssssssssssssssssssssssssssssssssssss. .$$$$$s.
   `$$.  `$$.   .ssss. .ssss. .ssss. .sss. .s  ss .sss. .ssss.    .$$'  .$$'
    `$$   `$$   ~ss.`$ $. ``$ $' .$$ $. `$ $$.s~' $. `$ $. ``$    $$'   $$'
  ds.`$ ds.`$     $$ ' $$ss~' $ss~$$ $$    $$S    $$~   $$ss~'    $'.sb $'.sb
 $$$$.$ $$$.$     $$   $$`~s. $   $$ $$..$ $$`~s. $$..$ $$`~s.    $.$$$ $.$$$$
 Y$$$$'.$$$$'.cC! ~~   ~~  ~~ ~   ~~ `~~~' ~~  ~~ `~~~' ~~  ~~    `$$$$.`$$$$Y
  `~'   `~'  `~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~'  `~'   `~'
```

User's Manual

** Please note that this document hasn't been udpated in a long time.
** Some information may be missing, which may be present in supplementary
** files such as DRIVERS.TXT, FILTERS.TXT, MIDI.TXT, UPDATE.TXT, etc.

1. Introduction

If you are familiar with tracker programs, then you could probably skip through most of this document. (I strongly suggest that ANYONE print out SUMMARY.TXT though, for a list of effects and editing commands.) For those of you who have had experience with Scream Tracker 3, read section 2.7. This should detail most of the major differences between Scream Tracker 3 and Impulse Tracker. For those of you who are unfamiliar with tracker programs it would probably be best to print out this document, and then refer to it as you work with the program.

Check out the file UPDATE.TXT. If this document contains conflicting information with what's listed in UPDATE.TXT, then follow what UPDATE.TXT contains, as I may have forgotten to update this text.

1.1 What is Impulse Tracker?

Impulse Tracker is a program used to create high quality music without the requirements of specialised, expensive equipment. The hardware requirements should be easily met: Any IBM 386+ compatible computer (although a 486+ is recommended), and a VGA or higher video card. If you want to hear sound, you'll need one of the following:

- Sound Blaster series (or most compatibles)
- Pro Audio Spectrum
- Windows Sound System
- ESS ES1868 AudioDrive
- Ensoniq SoundscapeVIVO
- EWS64XL Soundcards
- Gravis UltraSound
- Interwave based board
- Generic MPU401 for MIDI Input/Output
- DAC on LPT1/LPT2
- PC Speaker

Note that stereo effects (including surround sound) can only be experienced on software mixed stereo cards. Hardware mixed soundcards support panning, but do not support surround sound.

You'll need about 500k of conventional memory to get the program running - about 600k to have it load almost any song (as long as you have an EMM driver - samples are stored in EMS when possible, otherwise the songs that you are able to load will be limited even further!). If you use EMM386.EXE, add "H=255" at the end of it to prevent "Out of memory" messages when you still have EMS remaining. If you're using some other memory manager, check it's documentation to see how to increase the number of "memory handles".

Impulse Tracker supports direct control over 64 channels, and can load and play the following formats: S3M, MTM, MOD and of course, IT. At the moment, modules can only be saved in the IT and S3M formats.

The number of channels playable on hardware mixed soundcards is limited by the hardware. Check DRIVERS.TXT for specific details on your soundcard hardware.

The Gravis UltraSound can only play 32 channels, although this shouldn't be too much of a limitation. Note that if you put notes in channels 33->64 in sample mode using a Gravis, you WILL NOT hear these notes! (There's an explanation of sample mode much later on, just keep it in mind if you own a Gravis UltraSound - it shouldn't cause any problems.) The same restriction applies for the Sound Blaster AWE 32, but to 30 channels. (ie. notes in channels 31->64 will not cause any sound)

1.2 About Impulse Tracker.

Impulse Tracker began simply as an extension to Scream Tracker 3 (which should be quite obvious to Scream Tracker 3 users, due to the interface.) At first, I only intended a couple of extra features (eg. proper panning and a couple of other interesting functions), but with the release of Fast Tracker 2, it became obvious that there were many areas in which Scream Tracker could be improved. In spite of all the limitations of Scream Tracker 3 (in comparison to Fast Tracker 2, which offered Volume/Panning Envelopes, 16-bit samples, Samples > 64k, an in built sampler, proper stereo panning on SB16, etc. etc), there were still more .S3M releases than there were .XM releases! And the reason must have been within the way you WRITE music with Scream Tracker 3. That's why I've used the same simple and QUICK interface that Scream Tracker 3 offered. And I've also incorporated all the power of Fast Tracker 2 and more!

The Tracker was written in 100% Assembler. All the routines are of my own coding (That's why some of them suck so much :)).
The program was written on and off during 1995, but most of the work was completed in the summer holidays of '96. The source code is over 100,000 lines long and occupies over 3MB. The Tracker runs entirely in text mode (!) with some neat remapping of characters (that's why I haven't been able to use colours to help in some places ie. making the volume envelope nodes a different colour would have made it easier to use!).

1.3 Running Impulse Tracker.

Win95 Users - ignore all this HIMEM and EMM386 stuff.. as long as you don't have "noems" or "noframe" as a parameter to EMM386 in your config.sys file, it'll work fine. If you have no idea what I'm talking about, you can probably assume it's fine :) (unless you get "out of memory messages")

Impulse Tracker uses EMS. If you want to be able to load large songs, you need the following lines in your CONFIG.SYS file (on your boot drive)

```
DEVICE=<path>\HIMEM.SYS
DEVICE=<path>\EMM386.EXE RAM H=255
```

eg. if the files are in your C:\DOS directory, you need this:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM H=255
```

If you use QEMM, use the following line instead:

```
DEVICE=<path>\QEMM.SYS DMA=64, HANDLES=255
```

If you want to run Impulse Tracker in Windows 95, check out WIN95.TXT

To find out details about the driver for your soundcard, check through DRIVERS.TXT.

Most users should not require any command line switches, however the following are available:

```
SFilename.Drv - Sets sound driver to use. eg. IT /sITIW.DRV
                this may become necessary as new sound drivers
                are released.

S#       Set sound card
          S0 = No sound card (silent mode)
          S1 = PC Speaker
          S2 = Sound Blaster
          S3 = Sound Blaster 2
          S4 = Sound Blaster Pro
          S5 = Sound Blaster 16
          S6 = Sound Blaster AWE 32
          S7 = Gravis UltraSound
          S8 = Interwave
          S9 = Pro Audio Spectrum
          S10 = Pro Audio Spectrum 16
          S11 = Windows Sound System
          S20 = .WAV writer device

         If this parameter is omitted, then a (hardware) detection
         routine is used. Check DRIVERS.TXT for specific information
         on the various drivers.

Axxx     Set Base Address of sound card (hex)
D#       Set DMA of sound card (decimal)
I##      Set IRQ of sound card (decimal)

M#####   Set Mixspeed. Values are accepted between 0->65535,
         however, the soundcards have hardware limits which will
         override the command line switch. Again, check DRIVERS.TXT
         for specific information on your sound driver.

L###     Limit number of active channels

         When you limit the number of channels, you are limiting the
```

number of notes that you can hear on playback. In a "Sample" controlled song (explained later), the result is that if you play a 16 channel piece with the limit at 4 channels, then you will lose the last 12 channels! Even if the first 4 are not playing anything. In an instrument controlled song, the result is that you will hear 4 notes at most, which can be controlled from any of the 64 channels, excess notes will be lost. (The reason for this is within the channel allocation routines.)

Note that the hardware mixed devices (eg. Gravis UltraSound, AMD Interwave and Sound Blaster AWE32) cannot play more than a certain number of channels (drivers.TXT will provide the exact details).

If you are new to tracking and didn't understand what I was just talking about, hilight this section with a marker (you've printed this out, right?), and in your dabblings, if you notice that some notes aren't being played, reread this section. It should make sense by then, and the reason hopefully lies in the above paragraphs.

V1 Override VGA detection routine.. in case you DO have a VGA and the program doesn't recognise it. If characters look weird, you may need to use this switch to override the Matrox mode autodetect.

V2 Force Matrox mode. Use this switch if you get garbled stuff appearing on your screen

R Reverse stereo channels.
This is only really useful on the SB16. (Swaps left/right outputs). The Sound Blaster Pro stereo setting routines aren't accurate, and will sometimes set the left to left and right to right (how it's supposed to be), and sometimes set the left to right and right to left (how it's not supposed to be :)).
You can also switch left/right channels in IT by pressing Alt-R on the info page.

C Control playback in DOS Shell.
When this option is on, the following keys will operate within the DOS Shell:

Right-Alt:	Stop playback
Right-Ctrl:	Play song (if not already playing)
Grey Plus:	Increase global volume
Grey Minus:	Decrease global volume

Warning: There are problems on some computers with this enabled! If your keboard locks up, it may be necessary to press the left ctrl/alt to "unfreeze" it... sometimes (other times.. I don't know!)

F Disable file colour distinctions.
When this is on, all file colours will appear the same as the background colour. This is just to make it easier to redo the palette to your liking more easily.

1.4 Technical Information

Formats supported

Modules:

MOD (M.K., M!K!, 4CHN, 6CHN, 8CHN, xxCH, FLT4, FLT8)
669 (Composer 669, Unis669)
MTM (MMEdit files)
S3M (Scream Tracker 3 files)
XM (Fast Tracker 2 files, DigiTracker 3 files)
IT (Impulse Tracker 1.xx, 2.xx files)

Samples:

S3I (Scream tracker sample, 8 bit or 16 bit)
IFF (Fast tracker 2 sample, 8 bit or 16 bit)
WAV (Microsoft WAV, 8 bit or 16 bit)
Wxx (TX Wave format samples)
ITS (Impulse tracker sample, 8 bit or 16 bit)
RAW (Raw sample information, assumed 8 bit)
PAT (Gravis UltraSound patches)
KRZ (Kurzweil Synth files)
MOD (Samples from almost all .MOD formats)
PTM (Samples from Poly Tracker modules)
669 (Samples from 669 and 669 Enhanced modules)
FAR (Samples from Farandole composer modules)

```
    MTM (Samples from MMEdit modules)
    S3M (Samples from Scream Tracker 3 modules)
    XM  (Samples from Fast Tracker 2 modules)
    IT  (Samples from Impulse Tracker 1.xx, 2.xx modules)

  Instruments:
    XI  (Instruments from Fast Tracker 2)
    ITI (Instruments from Impulse Tracker)
    XM  (Instruments from Fast Tracker 2 modules)
    IT  (Instruments from Impulse Tracker 1.xx, 2.xx modules)

Modules
  99 Samples maximum
  99 Instruments maximum
 200 Patterns maximum (from 32-200 rows per pattern)
 256 Orders maximum
  64 Channels under direct control
 256 Channels maximum for virtual control (64 max default)

Samples
  8/16 bit samples
  Maximum size around 4MB
  No Loop/Forwards/Ping pong loop
  Default pan for samples (optional)
  Vibrato parameters

Instruments
  25-point Volume/Panning/Pitch envelopes
  Default pan for instruments (optional)
  Pitch pan separation
  Volume / panning swing
  Virtual controls (*REALLY* worth understanding)
  MIDI Output controls

Sound Devices
  Generic MPU401
  Sound Blaster 1.xx, 2.xx (8 bit mono)
  Sound Blaster Pro (8 bit stereo)
  Sound Blaster 16 (16 bit stereo, MIDI In supported)
  Sound Blaster AWE 32 (16 bit stereo, hardware mixing, MIDI In &
                        MIDI Out supported)
  Pro Audio Spectrum (8 bit stereo)
  Pro Audio Spectrum 16 (16 bit stereo)
  Windows Sound System card (16 bit stereo)
  ESS ES1688 AudioDrive (16 bit stereo)
  ESS ES1868 AudioDrive (16 bit stereo, MIDI In supported)
  ESS ES1869 AudioDrive (16 bit stereo)
  Gravis UltraSound (16 bit stereo, hardware mixing)
  Interwave based board (eg. GUS PnP, Dynasonix, 16 bit stereo,
                             hardware mixing, MIDI In & MIDI Out
                             supported)
  EWS64XL (Codec driver, 16 bit stereo)
  Sound Track '97 PCI (16 bit stereo)
  DAC on LPT1/LPT2 (8 bit mono)
  PC Speaker (very final option not recommended... )

MIDI Output
  Supports 128 MIDI Macros
  Supports 16 parmaeterised MIDI Macros
  Internal handling of conflicting notes
  Fully configurable output messages
```

2. Using Impulse Tracker

Before any specific details of how to use the tracker, it may be worth knowing these few "editing" keys:

When using Thumbbars.

Pressing Left and Right arrows will shift the values left/right
Holding down Ctrl while pressing left/right will move them quicker
And Shift-Left/Right will move them even quicker....
But just typing in a number will get you directly to the desired value!

String (text) entry.

Nothing special here (and the routines need to be rewritten), but pressing Ctrl-Backspace will clear the entire text..

Numerical entries (of the 7 digit or 3 digit variety)

Pressing '+' or '-' will increase/decrease the value by 1. This is especially useful when modifying loop values.

2.1 Playing songs

If you are like most of us, you would have run the program first, and probably figured out how to do this yourself. Songs are simply loaded by pressing F9 at any time (to invoke the file load menu) and then selecting a song by pressing Enter. You can also type the first few characters of a filename, and a "search" will be made for the first match.

Note that on all the file menus in Impulse Tracker, you can also delete files by pressing Delete.

After loading a song, use F5 to play it and F8 to stop. The Info Page should appear, and you can cycle through views using PgUp/PgDn. Separate view windows can be created by pressing Insert (and removed by pressing delete) and these separate windows can each have a different view method.

Details (ranges follow in brackets):

Frequency - the speed (pitch) at which the sample is played in samples per second
Position - the offset in a sample (in bytes)
Smp - The sample currently being played (1->99)
FVl - The final volume of the sample, taking into account all the scaling factors (viz global volume, sample vol, envelope volume, channel volume and fadeout) (0->128)
CV - Channel volume (0->64)
SV - Sample volume (0->64)
VE - Envelope volume (0->64)
Fde - Fadeout component (0->512)
Pn - Panning (0->64, Su = surround)
PE - Panning envelope value (0->32)
NNA - The current NNA (Cut/Con/Off/Fde)
Tot - The total number of active virtual channels 'owned' by the channel

If you're looking for some songs to play, check out the following places:

ftp.cdrom.com/pub/demos/music [huge collection!]
kosmic.wit.com/kosmic/songs
ftp.uni-muenster.de/pub/sounds
archie.au/pub/aminet/mods

2.2 Pattern editor (F2)

The pattern editor allows you to edit patterns. For those of you who are familiar with music, patterns can be thought of as "bars", and the order in which these "bars" are played is determined by order list. For those of you who are not familiar with music, consider patterns as a small collection of notes. Impulse Tracker supports up to 200 different patterns - it *IS* quite a lot more than it may initially sound!

Each pattern can range in length between 32 and 200 rows. You can change this value by using the "Pattern Editor Configuration" screen by pressing F2 when alredy in the Pattern Editor. (If you want to change the number of rows of several consecutive patterns, use Ctrl-F2)
The other options available are the base octave (explained later), the skip value (also explained later), the row hilight major and minor (which determines the distance between the emphasized rows) and the command/commandvalue link/split option, which determines whether when editing, the cursor should move downwards when entering an effect, or across to the effect value columns.

The pattern editor appears normally as 5 'channel' columns as such:
(You can configure the pattern editor .. press F1 in the pattern

editor, then page down to find the Track View options.)

```
   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~L
   ~C-5 01 23 A02~E-5 02 64 D01~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~
   ~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~
   ~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~
   ~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~
   ~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~
   ~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~~~~ ~~ ~~ .00~
                   ^   ^  ^  ^
Component Columns: 1   2  3  4
```

What the component columns mean:

1) The first column contains the note and octave of the note. Notes are entered by using the keyboard as such:

```
(Note)         C# D#    F# G# A#    C# D#    F# G# A#    C# D#
            ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~
            ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~
(What you   ~ ~S~~D~ ~ ~G~~H~~J~ ~ ~2~~3~ ~ ~5~~6~~7~ ~ ~9~~0~ ~
 type)      ~ ~~~~~~ ~ ~~~~~~~~~ ~ ~~~~~~ ~ ~~~~~~~~~ ~ ~~~~~~ ~
            ~ Z~ X~ C~ V~ B~ N~ M~ Q~ W~ E~ R~ T~ Y~ U~ I~ O~ P~
            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

(Note)        C  D  E  F  G  A  B  C  D  E  F  G  A  B  C  D  E
             (Octave 0)           (Octave 1)           (Octave 2)
```

(For those of you with AZERTY keyboards, you will find that the keys should work as positioned on the keyboard, rather than having to figure out QWERTY equivalents)

The octave of the note is determined by adding the BaseOctave to the Octave of the note played. The BaseOctave can be adjusted by pressing the Grey keys '/' or '*' or using Ctrl-Up Arrow or Ctrl-Down Arrow.

The range of notes is from C-0 to B-9. The 'middle' note is considered as C-5

Pressing '1' on the note column will enter a notecut command. This causes any note in the column to immediately stop.

Pressing '`' (the note below ESC) on the note colume will enter a noteoff command. This causes all sustain points to be released (Explained in samples and instruments in more detail.)

At the bottom of the channel is an 'edit mask'. This highlights what will be affected when you type anything in. Note that typing a note in can affect more than just a note - normally it's set to enter an instrument and volume with it! You can easily tell IT to enter an effect with it also by changing the channel mask with ',' (comma) on columns 2, 3 and 4.

Examples of interpretation:

```
~C-4 01 ~~ .~~ - will play note C octave 4, instrument 1
~D-4 ~~ ~~ .~~ - will play note D, octave 4, instrument 1
~~~~ 02 ~~ .~~ - Will play note D, octave 4, instrument 2
~E-4 ~~ ~~ .~~ - will play note E, octave 4, instrument 2
~G-6 12 ~~ .~~ - will play note G, octave 6, instrument 12
```

2) The second column contains the sample/instrument, depending on whether the tracker is operating in sample mode, or instrument mode. Valid ranges are from 01->99 (decimal).

3) The third column contains the volume or panning of the note. To toggle between entry of volumes and panning, press '`'

Volumes.
0 is the softest (ie. nothing) and 64 (decimal!) is the loudest. If no volume is specified, then the default volume for the sample is used. (Note that effects *CANNOT* raise the volume above 64)

The volume scale works linearly - ie. one note played at a volume of 64 will be the same 'loudness' as 4 of the same note, one played at a volume of 10, another at a volume of 30, another at a volume of 20 and another at a volume of 4.
 (10+30+20+4 = 64)
But if you have a choice, always go for the single note (it will require less processing (occupy less channels), and also sound better in quality, due to roundings that occur in processing softer volumes).

Panning.

Panning controls appear in a different colour to the volume controls. For panning in this column, 0 represents far left, and 64 represents far right. If you have an Xxx effect at the same time, the Xxx will take precedence over this column.

It is more efficient filesize-wise to use a panning control in this column rather than in the effect column.

Volume, Panning & Effect??
OK. So you want to do something tricky. Well, it's possible to get all 3 of these how you want with a little meddling with the channel volume control - just place one before the row with an appropriate value (remember that the range is 0->40 HEX) such that the channel volume scales the default volume to the desired value.... What I mean is that if you want to play a note at volume 32, pan 48, with vibrato H81, then you can do the following:

```
~~~~ ~~ ~~ M20~ - Set channel volume to HALF.
~C-4 01 48 H81~ - The 48 is a pan command - this assumes that
                  the default volume for sample/instrument 1
                  is 64.
```

Volume Column Effects
In IT208 and higher, some extra functions are available in the volume column. These allow you to slide the volume up/down and pitch up/down, just like the final column effects. If you aren't familiar with the rest of the editor, leave this section out for now and come back after you have become acquainted to column (4) of the editor - the effects column.

Volume column effects are selected by pressing A-H in the first column of the effects.

```
Ax = fine volume slide up by x
Bx = fine volume slide down by x
Cx = volume slide up by x
Dx = volume slide down by x
Ex = pitch slide down by x
Fx = pitch slide up by x
Gx = portamento to note with speed x
Hx = vibrato with depth x
```

In all cases, if x is 0, then the effect memory is used (as explained in (4)

The memory for Ax/Bx/Cx/Dx are shared, as is the memory for Ex/Fx.

4) The final column contains effect data. For those of you who are just starting, I would advise you to leave this section until later, when you have already dabbled with entering notes, and want some special features. It's easy to get caught up with all these special features and you can 'overdo' the effects. And it will sound absolutely pathetic.

Trust me, I know - I've done it :)

Effects are entered by typing 'a'-'z', then a hex value (see later for an explanation of hex numbers) as the final two entries. The effects allow you to a wide variety of functions that are otherwise impossible to obtain.

Due to the repetitive nature of some effects, there is a 'memory' so that instead of typing:

```
~~~ ~~ ~~ G12     It is easier to    ~~~ ~~ ~~ G12
~~~ ~~ ~~ G12     use:               ~~~ ~~ ~~ G00
~~~ ~~ ~~ G12                        ~~~ ~~ ~~ G00
~~~ ~~ ~~ G12                        ~~~ ~~ ~~ G00
~~~ ~~ ~~ G12                        ~~~ ~~ ~~ G00
~~~ ~~ ~~ G12                        ~~~ ~~ ~~ G00
~~~ ~~ ~~ G12                        ~~~ ~~ ~~ G00
```

The following effects 'memorise' their previous values:
(D/K/L), (E/F/G), (HU), I, J, N, O, S, T, W

Note: Bracketed commands share the same 'memory' value. So

```
~~~ ~~ ~~ E12   can be written as:  ~~~ ~~ ~~ E12
~~~ ~~ ~~ F12                       ~~~ ~~ ~~ F00
~~~ ~~ ~~ E12                       ~~~ ~~ ~~ E00
~~~ ~~ ~~ F12                       ~~~ ~~ ~~ F00
C-4 01 ~~ G12                       C-4 01 ~~ G00
```

Commands H and U are linked even more closely.
If you use H00 or U00, then the previous vibrato, no matter whether it was set with Hxx or Uxx will be used. So:

```
~~~ ~~ ~~ H81    Is the same as:     ~~~ ~~ ~~ H81
~~~ ~~ ~~ U00                        ~~~ ~~ ~~ H81
~~~ ~~ ~~ U83                        ~~~ ~~ ~~ U83
~~~ ~~ ~~ U00                        ~~~ ~~ ~~ U83
~~~ ~~ ~~ H00                        ~~~ ~~ ~~ U83
```

Hex Numbers

Note: Impulse Tracker works ENTIRELY with decimal numbers EXCEPT for the effects column.

Instead of using a decimal system (ie. base 10), it is more natural for the computer to work with hexadecimal (often abbreviated to simply 'Hex') - numbers which operate in base 16. The first 9 numbers in hex are denoted by '1' to '9' and the next 6 are denoted by 'A' to 'F'. So if you count in hex, it will be as follows: (0), 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 2A etc.

To convert a hex number to decimal, multiply the 'tens' column by 16 and add the value of the second column. ie. 32 Hex = 3*16+2 = 50 decimal. 2A hex = 2*16+10 = 42. (because A = 10) The maximum number that you can represent with two Hex digits is FF = 255 decimal.

Hope that this makes SOME sense :)

Effects.

Axx Set Speed.

I prefer to think of this command as "Set Frames per Row". Normally, the tracker operates at around 50 frames a second. If the rows were played at this speed, then a huge amount of space would be required to enter the pattern data. Instead, setting the 'speed' of the song will cause the tracker to wait on the current row for 'xx' frames. Hence, setting the speed at 50 (decimal = 32hex) will cause each row to last about a second - quite a long time! The default is A06. The initial speed can be set in the variables screen on F12.

If two Axx commands are given in the same row, then the command in the higher channel (by number) will take effect.

Bxx Jump to order.

Causes the song to jump to order xx (hex). This is often used to create looping songs.

If two Bxx commands are given, then the command in the higher channel will take effect.

Cxx Break to row.

The Command Cxx signifies the end of the current pattern, and also that the next pattern should be played from row xx (hex)

If two Cxx commands are given... you know :)

D0x Volume slide down

The volume slide down command causes the volume of the note to be reduced by x for each frame after (for that row). ie. If you have a note at volume 64, with command D01 and speed A06, then the final volume will be 64-5=59. A note at volume 32 with command D02 at speed A05 will result in a volume of 24.

For greater S3M compatibility, D0F will cause the volume to drop by 15 EVERY frame, instead of just off-note frames.

Here's a full frame-by-frame description which should provide greater understanding:

```
At 'speed' 4:
 C-4 01 .. D04
```

What this does is:
Frame 1: Plays C-4 with instrument 1
Frame 2: Lowers the volume by 4
Frame 3: Lowers the volume by 4
Frame 4: Lowers the volume by 4

The next frame will be controlled by the next row of information.

Dx0   Volume slide up.

Operates exactly the same way as D0x, but slides the volume up by 'x'. Volumes cannot exceed 64 (checked and limited internally).

For greater S3M compatibility, DF0 will cause the volume to rise by 15 EVERY frame, instead of just off-note frames.

DFx   Fine volume slide down.

Takes the volume down by x at the start of the row.

DxF   Fine volume slide up.

Takes the volume up by x at the start of the row.

Exx   Pitch slide down

The pitch will slide down with speed xx (hex). In linear frequency mode, a pitch slide down by a particular value will always cause the same "pitch interval" - this does not occur in Amiga frequency mode. Valid ranges for xx are between 0 and 0DFh (as > 0E0h will be interpreted as fine slides)

EFx   Fine pitch slide down

Slides the pitch down by x at the start of the row.

EEx   Extra fine pitch slide down

Same as EFx, but 4 times finer. (ie. EE4 is equivalent to EF1)

Fxx   Pitch Slide up
FFx   Fine Pitch slide up
FEx   Extra fine pitch slide up

Operates in the same manner as the Exx commands, but slides the pitch up. If the pitch gets 'too high', then the channel is turned off.

Gxx   Portamento to note.

This command requires 2 parameters: A note to slide to and a speed. Example:

```
C-4 01 ~~ .00
G-4 01 ~~ G08
~~~ ~~ ~~ G00
~~~ ~~ ~~ G00
```

This will cause the note C-4 to slide to G-4 with speed 8.

You *CAN* change the sample that the note is sliding to. It is a good idea to make sure that the C5Speed of the samples is similar, if you want to try this! The new sample will play from it's beginning.

Hxy   Vibrato with speed x, depth y

Causes the frequency to osciallte with depth 'y' at speed 'x' which causes an interesting effect. Don't get carried away with it though! If you specify EITHER x or y as 0, then the previous value will be used.

Ixy   Tremor with ontime x, offtime y.

Causes the volume of the instrument to remain normal for x frames, then sets the volume to 0 for y frames.

Jxy   Arpeggio with halftones x, y.

This causes the note to quickly cycle through three notes - the note playing, a note x halftones above, and a note y halftones above. This causes an effect similar to old C-64 chords.

Example: C-4 01 ~~ J47 will cause the notes C-4, E-4 and G-4 to be cycled.

Kxx   Dual command: Vibrato + Dxx (Volume slide)

Note: The vibrato could have been set with either Hxx or Uxx

Lxx   Dual Command: G00 + Dxx (Portamento to and Volume slide)

Mxx   Set channel volume

Each channel has a volume range from 0->40h. The lower the value, the softer the notes in the channel. This command is convenient for making 'echoes'... you won't have to modify the echoing channel from the original, except for placing a M20 at the top of it!

Values greater than 40h are ignored at playtime.

N0x, Nx0, NFx, NxF Slide channel volume commands

These commands work in the same manner as the slide volume commands, but operate on the channel volume, rather than directly on the note volume.

Oxx   Set sample offset.

This will cause a sample to be played from offset yxx00h. This is useful to skip past the first part of a sample which may have a loud hit, or to start a speech sample half way through.

If you specify a value PAST the end of a sample, then the command is ignored.

The "y" part of yxx00h is set with command "SAy"

Example:

```
... .. .. SA5
C-4 01 .. O10
```

This will play note C-4, instrument 1 at offset 51000hex

Note to programmers: Oxx for 16-bit samples will move to the xx00h*2 position in the sample - ie. the 'xx00th' sample.

P0x, Px0, PFx, PxF Slide panning commands.

These work in the same maner as the slide volume commands, but operate on the channel panning. P0x slides the panning right, while Px0 slides the panning left.

Qxy   Retriggers a note after y frames with volume modifier x.

This will cause a sample to be replayed from it's beginning after every y frames. 'x' can be any of the following:

Value	Effect on volume each retrig.
0	No change
1	-1
2	-2
3	-4
4	-8
5	-16
6	*2/3
7	*1/2
8	No change
9	+1
A	+2
B	+4
C	+8
D	+16
E	*3/2
F	*2

Rxy   Tremelo with speed x, depth y

This command acts similarly to the vibrato command, but affects the note volume, instead of the pitch. If you specify EITHER x or y as 0, then the previous value will be used.

S3x   Sets vibrato waveform to x
S4x   Sets tremelo waveform to x
S5x   Sets panbrello waveform to x

```
Values for x: 0 = sine wave
              1 = square wave
              2 = Ramp down
              3 = Random!
```

S70   Past note cut
S71   Past note off
S72   Past note fade

With the loss of 1-1 correspondence of editing channels to internal playing channels, these commands provide control over notes that have already been played by a channel.

Example:
Consider the following situation, where instrument 1, notes C-4 & D-4 map to a looped sample, and the New Note Action is set to continue.

```
C-4 01 64 .~~
D-4 01 64 .~~
E-4 01 64 .~~ <--- Use S70 here!
```

Ordinarily, control would be lost over C-4 and D-4, but using command S70 at the point indicated, the notes C-4 and D-4 would be stopped immediately when the note E-4 plays.

Similarly, note off commands and Fadeout commands can also be issued.

Note: You *MUST* be in instrument mode for these to have any significance.

S6x   Pattern delay for x ticks.

S73   Set NNA to note cut
S74   Set NNA to continue
S75   Set NNA to note off
S76   Set NNA to note fade

These options allow you to override the default NNA for a particular NOTE (ie. it does not affect the instrument) See section 2.5 for an explanation of NNAs.

Note: You *MUST* be in instrument mode for these to have any effect.

S77   Turn Volume envelope off.

Stops the processing of the volume envelope. This is for the times that you don't want to use the volume envelope that you've created.

S78   Turn volume envelope on.

Sometimes, you'll decide that it's better not to use the volume envelope for most cases, but you may just want to use it a few times - this is the command that lets you do that!

Note: You *MUST* be in instrument mode for this to work.

S8x   Set panning position

Set panning position, S80 is leftmost, S8F is rightmost. This is an obsolete command, provided only for ST3 compatibility. Use command Xxx instead.

S91   Set surround sound!

For those of you who are lucky enough to have a dolby surround sound decoder (myself not included - so if ya wanna send me something.....), this *should* cause the sound to come from the surround speaker. This is a pseudo-panning command, and issuing either Xxx or S8x will cause the surround sound to be reset.

If you try and play Surround Sound on a GUS, Interwave or AWE32, it will be played as a central pan.

SAy   Set high-offset. Check Oxx for an example.

SB0   Set loopback point

SBx   Loop x times.

This pattern space-saving feature will cause the pattern to be looped x times back to the last SB0. Note that you can only loop within the pattern! Also, each channel has it's own loop-back information... so you HAVE to have the SB0 and SBx in the same channel for it to operate.

SCx   Note cut after x frames

This will cause a note to be immediately stopped after x frames. It is similar to issuing a "^^^" in the note column, just that it has a finer control over timing.

SDx   Note delay for x frames

Actually, this command works like "interpretation" delay for x frames. Any data - note, sample/instrument or volume in the channel will not be interpreted until x frames into the row.

SEx   Pattern delay for x rows

This will cause a "pause" on the row for effectively x rows longer.

Note that if TWO pattern delay commands are issued, then the only the command in the higher channel will be considered.

Txx   Set tempo to xx

Valid ranges are between 20h and 0FFh. The higher the value, the faster the playback. This essentially determines the time length of each frame, by the following formulas:

Frames per minute = 24*Tempo

equivalently:

Frames per second = 0.4*Tempo

T0x   Tempo slide down

T1x   Tempo slide up

Tempo slides up and down are used to smoothly modify the speed of the song. The tempo is modified by ~x every non-row frame.

Uxy   Fine vibrato with speed x, depth y

Same as vibrato, but 4 times finer in depth.

Vxx   Set Global volume

Valid ranges are between 0 and 80h. All notes playing are affected by this change.
Values greater than 80h are ignored at playtime.

W0x, Wx0, WFx, WxF Slide global volume

Similar to the Dxx commands, but operate on the global volume. Useful when fading out songs.

Xxx   Set panning position

Sets the panning position anywhere from left to right (X00 is left, XFF is right).

Yxy   Panbrello with speed x, depth y.

What the hell is panbrello? Well, it's a word that I coined late one night after a friend suggested that there should be a random pan position command. I decided.. why not - I'll make the vibrato/tremelo equivalent of panning! And so you have here, 'panbrello.' What this does is instead of oscillating frequency (pitch) or volume, it oscillates the panning position about the 'set' panning position for a channel. This is for you techno freaks out there who want another function to stuff around with! :)

The random pan position can be achieved by setting the waveform to 3 (ie. a S53 command). In this case *ONLY*, the speed actually is interpreted as a delay in frames before another random value is found. so S14 will be a very QUICK panbrello, and S44 will be a slower panbrello. With any other waveform, the higher the value for x, the faster the panbrello, like vibrato and tremelo. If you don't know what I'm talking about, check out the stereo-indicators on the info page (don't forget that you have to be in stereo mode!)

If you specify EITHER x or y as 0, then the previous value will be used.

5) Editing Functions

For a complete list of available functions, check the help screen for the pattern editor (on F1)

Ctrl-Backspace	IMPORTANT! This one will save you a lot of frustration - it's a 10-stage listed Undo function!

Grey Plus	Advance to next pattern
Grey Minus	Goto previous pattern
Shift-GPlus	Advance 4 patterns
Shift-GMinus	Go back 4 patterns
Ctrl-GPlus	Go to the next order's pattern
Ctrl-GMinus	Go to the previous order's pattern
Alt-Delete	Remove an entire row from the pattern
Alt-Insert	Insert an entire row into the pattern
'.'	erase data.
Space	will enter the previous data for that column.
'4'	play the note under the cursor.
'8'	play entire row.
Ctrl-F6	play pattern from the current row.
Ctrl-F7	Set playback mark - this will be the position that subsequent playback will occur when you press F7. Remove the mark by pressing Ctrl-F7 on the row that is already 'marked.'
Alt-0 -> 9	Quick select "skip value" The skip value is the number of rows that the editor advances by when a note is entered - this makes it easy to enter notes on alternate rows (press alt-2 first!) If the skip value is 0, then the editor will advance to the next channel (making it easier to enter chords) If the skip value is 0, AND template mode is on, then entering a note will cause the editor to advance by the length of the template block.
Alt-Enter	Store current pattern in memory
Alt-Backspace	Restore current pattern

6) Block Functions

Alt-B	Mark top left of block
Alt-E	Mark bottom right of block
Alt-L	Mark entire column (channel)
Alt-L*2	Mark entire pattern
Alt-D	Mark the minor row hilight number of rows. Pressing this repeatedly doubles the length of the block. eg. Pressing Alt-D once may mark out 16 rows, pressing it twice will mark 16 rows.
Alt-A	Transpose all the notes in the block a semitone down If no block is marked, then the current note will be transposed a semitone down.
Alt-C	Copy marked block into the clipboard
Alt-F	Double the length of the selected block
Alt-G	Halve the length of the selected block
Alt-I	Toggle Template mode... read part 8

```
Alt-J        Volume Amplifier... modify the volume controls
             between 0 and 200%
Alt-K        Slide volume... if you want to manually control a
             fadeout, or fadein, then set the volumes at the
             extreme ends of the blocks, then press Alt-K ONCE!
Alt-K*2      Remove all volume controls in the selected block
Alt-M        Mix clipboard with pattern data.. try it out, and
             you'll understand (it's like an advanced paste
             command - it'll only copy the data IF there is nothing
             on the row it is copying to.)
             Pressing this two times will copy the data if there's
             nothing in the same FIELD (not just same row).
Alt-N        Toggle Multichannel mode.. Multichannel mode allows
             you to select for each channel whether it should be
             part of the multichannel list. If you enter a note
             while on a channel which is in the multichannel
             list, then the editor will advance to the next
             channel.
Alt-N*2      Multichannel menu.
Alt-O        Overwrite pattern data with clipboard. Like paste,
             but without the "insert" part :)
Alt-P        Paste clipboard into pattern at current position
Alt-Q        Transpose all the notes in the block a semitone up
             If no block is marked, then the current note will
             be transposed a semitone up.
Alt-S        Set all the instrument (sample) values to the
             current instrument (sample) within a block
Alt-U        Unmark block / release clipboard from memory.
Alt-V        Set all the volume controls to the current
Alt-W        Remove all volumes not associated with a note/inst
Alt-X        Slide command value. Similar to Alt-K (slide
             volume). This command can be convenient to slide
             the panning from one position to another, or to
             slide global volume effects, channel volume, sample
             offsets... whatever...

Alt-X*2      Erase all effect data in the selected block!
Alt-Y        Swap selected block with a samesized/shaped block
             starting at the current position. Swap blocks
             CANNOT overlap.
Alt-Z        Cut current block.
             Warning: If you don't have enough memory, the
             current block WILL be erased, although there is no
             record of it in the clipboard....
```

7) Track View functions

Sometimes you just want to know what you've put in those tracks that are currently off the screen... these functions allow you to do just that!

```
Alt-T        Cycle Track view mode.
Alt-R        Remove all track views
Alt-H        Remove track view divisions (allows you to view 36
             channels!)
Ctrl-0       Remove current track from track view
Ctrl-1->5    Quick select a track view for current channel.
```

8) Templates

Templates are an invention to make entering repeated 'sequences' easier. Often, you may want to enter a note with the same effects following each note. eg.

```
C-4 01 00 D10   }
~~~ ~~ ~~ D00   }
~~~ ~~ ~~ D00   } Enter this....
~~~ ~~ ~~ D00   }
~~~ ~~ ~~ D02   }
~~~ ~~ ~~ D00   }
D-4 01 00 D10     }
~~~ ~~ ~~ D00     }
~~~ ~~ ~~ D00     }
~~~ ~~ ~~ D00     }
~~~ ~~ ~~ D02     }
~~~ ~~ ~~ D00     } Use the templates to enter this!
E-4 01 00 D10     }
~~~ ~~ ~~ D00     }
~~~ ~~ ~~ D00     }
~~~ ~~ ~~ D00     }
~~~ ~~ ~~ D02     }
~~~ ~~ ~~ D00     }
etc..
```

With templates, this can be done by entering the first 'block', then marking it and copying it into the clipboard (using Alt-C). Toggle the template mode by pressing Alt-I to Template: Overwrite, then go to the start of the second block, and press the key for D-4 (normally X). The whole block will be filled in! This is easiest to understand if you print this section out and try it in the editor.

The templates can also include other notes, which will be translated accordingly, and can span several channels.

A VERY neat use of templates is to enter multiple notes: (eg. octaves, different samples, left/right stereo sample pairs...)

```
    C-4 01 32 .00    C-5 02 32 .00  } Enter this...
    G-4 01 32 .00    G-5 02 32 .00      }
    D#4 01 32 .00    D#5 02 32 .00      }
    C-4 01 32 .00    C-5 02 32 .00      } And use Templates here!
    G-4 01 32 .00    G-5 02 32 .00      }
    D#4 01 32 .00    D#5 02 32 .00      }
```

Templates can span more than one row and one channel - you can template blocks of any size.

2.3 Order List, Channel panning & volume. (F11)

After creating patterns, the tracker has to know what order to play them back in. The order list is the means by which this is done. Another wording of the Order List which may make it simper to understand is the "Pattern sequence list." ie. the sequence in which the patterns are played back.
Simply enter the patterns into the list in the order they should be played, and voila! - you have a new piece of music. (as long as you have something in the patterns, of course!). The "---" marks the end of a song, and "+++" is simply a marker which is skipped (for ST3 support). Pressing N on a row will enter the previous order's pattern+1. eg. Entering 000, then pressing 'n' 5 times will enter 000 (which you typed), then 1, 2, 3, 4 and 5.

Initial channel panning and volume is the panning and volume that each channel is set to whenever the song is reset. It will not affect anything currently playing. On the panning list, you can also mute channels by pressing spacebar. Pressing S will select the initial 'panning' as surround sound. The initial channel volumes can be accessed by pressing F11 once you are already on the Order list and channel panning screen.

2.4 Samples

2.4.1 Information about Samples

Samples are the raw sound information. They can be of *anything* you like - normally a musical instrument of some sort, but you could have voice samples or sound effects too!

Impulse Tracker accepts 8-bit or 16-bit samples. It can import several different formats, which are detailed in section 1.4

If you import a 16-bit file which isn't recognised, you have to use the following steps:

1) Press Alt-A (to convert signed/unsigned) and convert the data.
2) Press Alt-Q (to toggle to 16-bit) and DON'T convert the data
3) If the sample is unsigned, you will then need to convert it AGAIN by pressing Alt-A (and selecting yes to convert data)

Samples have the following information: A name, a default volume, a global volume, vibrato information, loop information and a frequency

The default volume of a sample is the volume at which a sample is played, IF no volume is specifier.

The global volume is the modifier by which ALL occurrences of the current sample are scaled by.

Vibrato information - the vibrato speed is the speed at which the vibrato oscillates. Although it is possible to enter values 0->64, the best effects are obtained by entering values around 24-48. The vibrato depth determines the maximum deviation from the mean frequency. The vibrato rate determines how quickly the vibrato is applied. Higher values cause the vibrato to be applied more quickly, a low value like 1 can take many seconds before the vibrato is noticed.

Loop information. Samples can be looped to provide a continuous sound. The loop beginning and the loop end must be specified, and the loop method (either forwards or ping pong) must be set. (Toggle by pressing spacebar). The loop boundaries can also be modified using '+' or '-' (even while the sample is playing!)

Sustain loops are the loops used as long as no note off command is encountered. In the sample tester, releasing a key provides a note off command to the player. Once a note off command is issued, the normal loops apply. The idea behind sustain loops originated after listening to some synthesizers which provided 'after note' sounds - like fingers lifting off guitar strings. With sustain loops, you can have a guitar sound in the sustain loop, with the end of the sample containing the 'finger off' sound. This way, you should be able to achieve a new degree of realism. The best way to do this is to have a reasonably small sustain loop, and a section after it containing the note-off sound, possibly looped. (The sustain loop should ideally be less than 1/20th of a second, so that you have reasonably good control over the timing of the note-off, but this is not always possible)

Example: If you have a sample with no sample loop, but a sustain loop around bytes 19000-20000, then while no noteoff command is issued, then the sample will loop between 19000 and 20000. Once a noteoff command is received, then the sample will be played until it's end and then will stop.

If you have a sample of length 64000 with a ping pong loop around the whole sample (0->64000) and a sustain ping pong loop around bytes 20000-30000, then the sample will loop backwards and forwards between 20000 and 30000 until a noteoff command is received, after which it will loop between 0 and 64000 (backwards and forwards).

It is recommended that you do NOT have a ping pong sustain loop going to a forwards loop or no loop and that if you DO have a sustain loop AND a normal loop, that the sustain loop lies within the normal loop (for ping pong), or that the normal loop end is after the sustain loop end (for forward's loop) (think about what you're asking it to do in these cases carefully... it should make sense - the tracker can still handle the cases when this is not adhered to, but the resulting note-off effect can vary greatly depending on timing and pitch!)

The frequency of a sample determines its pitch. The higher the frequency, the higher the pitch. The frequency can be doubled or halved by pressing Alt-Grey Plus/Minus to cause an octave rise/decrease. For the technically inclined out there, the frequency is the number of bytes per second that have to be played for a C-5. You can increase the frequency by one semitone by pressing Ctrl-Grey Plus or decrease it by a semitone with Ctrl-Grey Minus.

To load in a sample, go to the entry in which you want to load it then press enter. The sample library should appear. You can test out any sample just by "playing" notes on the keyboard while you are on top of the sample. Load a sample into the sample list by pressing enter. Of course, if you don't have any samples, no samples will appear :)

Where can I get samples from??

You can steal ('rip') samples from modules by going to the sample in the sample list, then pressing Alt-O. Scream Tracker 3 samples can be saved by pressing Alt-T (vibrato, global volume + susloop information lost) or raw samples can be saved by pressing Alt-W (all variables lost!)

Warning: Be careful where you rip samples from!! Some people don't welcome it at all! And in some cases, sample ripping may be a breach of copyright - this mainly a concern when you use commercial samples)

There are also many great sample 'packs' available... try a local music BBS, or the bigger FTP sites.

Some synthesizers have patch files (eg. Kurzweil 2000 as .KRZ) which are available through the internet. You can download these, and convert them with an appropriate utility (I recommend Convert 1.4 - convrt14.zip on ftp.cdrom.com/pub/demos/programs/convert and use it to convert the files to .S3I format - even for 16 bit samples). I don't know what sort of copyright these samples have. *YOU* have to read any text accompanying the files to find this out.

Finally, you can make your OWN samples! With a sampler (eg. Digiplay 3.0) or with some other tracker (eg. Fast Tracker 2), you are able to record information through the microphone socket of the computer!

Note: The length of a sample is *NOT* necessarily the number of bytes it occupies! The length quoted is the quantity of SAMPLE INFORMATION. If you have an 8 bit sample, then the size in bytes IS the same as the length. If you have a 16-bit sample, then the size in bytes is DOUBLE the length.

2.4.2 Sample Functions

Alt-A and Alt-Q provide basic sample-conversion functions. Alt-A will convert a sample to or from unsigned to signed format. Alt-Q will allow you to change a sample between 8 and 16 bits.

If you want to remove the part of a sample before a loop or after the end of a loop, you can use Pre-loop cut sample (Alt-B) or Post-Loop cut sample (Alt-L)

To reverse a sample, use Alt-G. This can produce interesting effects.

If you are working to a size limit, check out the functions Alt-E and Alt-F. These functions allow you to resize the sample to whatever size you would like! (Note that there is a reduction in quality associated with a reduction in size). Alt-E will resize the samples WITH interpolation, Alt-F will resize the samples WITHOUT interpolation.

To decrease the volume of a sample, the best way for 8 bit samples is to reduce the Sample's GLOBAL volume. For 16-bit samples, it's better to just attenuate the sample (ie. use Alt-M to 75%). The sample will retain greater quality if you follow these guidelines.

Alt-M is used to amplify a sample to between 0 and 400%. The default value for Alt-M which appears is the maximum the sample can be modified without creating clipping distortions.

To exchange two samples *in the sample list only*, use Alt-X. To swap two samples *even in the pattern data*, use Alt-S.

2.5 Instruments

For those who have never used a tracker before, I strongly suggest that you skip this section for now - learn how to 'use' samples first. After that, you'll should be able to understand and hopefully fully appreciate the power that instruments can provide.

To enable instruments, go to the song variables (F12), and press on the "Instruments" button after then "Control" prompt. This *MUST* be done in order to use ANY instrument function (including special note effects, viz S7x).

The instrument parameters are split into four screens - one for each of general options, volume options, panning options and pitch options. To select the appropriate screen, just whack enter on any of the four buttons at the top of the instrument list.

Instruments are collections of samples (or just a single sample). The translation of samples is controlled by the column in the centre of the screen on the general options page which shows what note/sample pair a single instrument note will be translated to. Example: If you're on instrument 1, and the note translation table appears as such:

```
C-5~C-5 01
C#5~C-5 02
D-5~D-4 03
D#5~D#4 03
```

Then entering "C-5 01" into a pattern will cause sample 1 to be played at pitch C-5, "C#5 01", wil cause sample 2, to be played a pitch C-5, "D-5 01" will cause Sample 3 to play at D-4, "D#5 01" will cause Sample 3 pitch D#4 to play... get the idea?

Why would you want to combine several samples into one instrument? Well, one very good reason is that it makes it easier to enter drum parts - you no longer have to change the 'instrument' if you want another sample. And also, combining them into one instrument makes them easier to manage - you can specify a volume envelope, NNA or FadeOut which will apply for ALL of the samples in the list.

Another reason is if you want a very high quality sounding 'instrument.' If you wanted to go overboard, you could sample every single note on a piano separately, and set up the note translation table to point to each sample accordingly - you'd get a very, very nice sound - if you don't run out of memory first. Or patience :)

When you save an instrument to disk, all the related samples are stored with it. This means that you can setup a 'drum kit' instrument, and to load it into another song, you only need to select that instrument and all the samples will be loaded for you.

The Fadeout value for each instrument determines how quickly the volume of the instrument decays under any of the following conditions:

1) NNA "FadeOut" is selected, and another note is played. (see later in this section for info about NNAs)
2) The end of a volume envelope is reached.
3) A note off command is encountered, without a volume envelope.
4) A note off command is encountered, and the 'normal' volume envelope loop is on.

The larger the fadeout value, the quicker the volume decays.

Each instrument also has an associated volume/pan/pitch envelope! The volume envelope is editing by selecting the node (left/right arrows) then picking it up (with enter), moving it around (using arrows, or Alt-Arrows for quicker control), then pressing enter again to "put down the node". Nodes can be inserted or deleted (with the keys insert and delete!). Note that the envelope will only be used if the envelope flag is set to "on" (just below the envelope graph)..

Remember that you have to be in instrument mode for this all to work!

There are 3 pieces of information for each Envelope graph - the node number (hopefully obvious), the time of the node (the number of 'ticks' or 'frames' that elapse before the node point) and the value of a node (also hopefully obvious). So the x-axis is time, and the y-axis is volume/pan/pitch - simple, really!

You can also specify envelope loops and sustain loops. These operate in the same manner as the sample loops, but the numbers refer to node numbers.

The powerful feature of this tracker, though, is not the envelopes by themselves - it's the New Note Actions! What these options do is allow you to select what should happen to an instrument when another note is played in the same column. If NNA "Cut" is selected, then the previous note will immediately be stopped (like in all other trackers). If NNA "Continue" is selected, then the note will continue playing! This is especially useful for Drum Parts, where there may be a long-ish snare sample - you can go on ahead and put a bass drum in the very next row after it - the snare drum will still complete playing! NNA "Note Off" issues a note off command to a note when a new note is played in the channel. This is particularly useful in combination with volume envelopes and volume envelope sustain loops. NNA "Note Fade" causes the current note to fade out with the fadeout value when a new note is played.

WARNING!!

New Note Actions are EXTREMELY powerful, but they CAN cause problems if you are not careful. Selecting NNA Fade with a fade value of 0, or note continue with a looped sample (no volume envelope) or anything which can cause a quick build up of allocated channels can easily hang a slow computer (or even a quick one, for that matter!) I tried to put checks against this, but they always triggered too late - "past the point of no return" where the CPU becomes so bogged down with processing the information, that it can't do or try to do anything else. (The tracker has *MANY* calculations to do - the most time consuming are the mixing routines, which require the processor to process as many bytes as the mixing speed per second for EACH note ie. a mixing speed of 44kHz means that for each note playing, 44000 calculations have to be made EVERY second... so with 64 channels 'active' at the maximum mixing rate for a SB16, almost 3 MILLION calculations have to be done EVERY second to produce the sound (in mono)!!! (My 486 can cope with this, but my 386 just dies!)

Duplicate Check Type (DCT) / Duplicate Check Action (DCA)
   DCT = Off/Note/Sample/Instrument, DCA = Cut/Off/Fade

When the duplicate check type is enabled, then repetitions of the same instrument&note/sample (or just repititions of the same instrument) pair in a particular channel will cause the previous occurence of the instrument&note/sample pair to be cut or faded (depending on the DCA)

  Example: If the DCT is set to note, and DCA is set to Cut, then the
            asterixed notes will cut out the tilda notes

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└·
~  C-4 01 ~     ~     C-4 02 ~      ~   C-4 03 }                    ~
~  D-4 01       ~     C-4 02 *   ~  ~   D-4 03 } Nothing get's cut. ~
~  C-4 01 *     ~     C-4 02     *  ~   C-4 02 }                    ~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

This option was included to help limit the number of active channels, and is especially useful for drum tracks.

Another nifty application of Duplicity checks is the following example: You can have a separate sample for each string of a guitar and setup and instrument to accomodate this. Setting the NNA to continue, the DCT to sample and the DCA to fade (with a relatively quick fadeout) means that whenever you play a new note, the previous note will continue to play on. BUT! If a previous note of the same instrument has the same sample as the new note being played, then it will be faded out - this closely relates to what you hear when someone plays a guitar - when they play the same string, they have to put their fingers down on the string, which causes the last note *on that string* to fade out.

I sincerely hope that these options do not cause the death of brilliant 4 channel music - that would be a great shame! There is truly an art in making a decent sound in as few channels as possible! (If you're new to this sorta thing, then it'll grow upon you...)

For interest's sake...
Channels are turned off internally under any of the following conditions:

1) The end of a sample is reached (quite obvious)
2) When the end of a volume envelope is reached, and the final envelope volume is 0
3) When the fadeout value for a channel causes it to become silent.
4) When a duplicate note is played when DNT is set to Note and DCA is set to cut for the instrument.
5) When a notecut is issued (obvious)
6) When a channel is moved to the background (using NNAs) AND the volume is 0.

Out of all the functions provided for the Samples and Instruments, perhaps the only one that requires explanation is the "Update Pattern Data" function. This function was written for people who have already written music in MOD/S3M/MTM formats, and want to combine their percussion parts into a single instrument. The way to do this is to set up the instrument->note/sample table, and then use the "Update Pattern Data" function. What this does is search through all the patterns for all occurrences of the note/sample pairs that appear in the instrument->note/sample table, and replace it with the appropriate note/INSTRUMENT pair. Sounds quite complicated... sorry :)

Example:
  If you originally have the following setup

Sample 1 = Bass Drum
Sample 2 = Snare Drum
Sample 3 = Closed Hihat
Sample 4 = Open Hihat

  And the following column within any pattern:

```
C-5 01 ~~ .~~
C-5 03 ~~ .~~
C-5 03 ~~ .~~
C-5 03 ~~ .~~
C-5 02 ~~ .~~
C-5 04 ~~ .~~
```

```
C-5 01 ~~ .~~
C-5 04 ~~ .~~
C-5 01 ~~ .~~
```

You can combine these four samples into one instrument by creating the instrument->note/sample table with the following entries: (say for instrument 10)

```
C-5~C-5 01
C#5~C-5 ~~        <--- just an empty slot...
D-5~C-5 02
D#5~C-5 ~~
E-5~C-5 03
F-5~C-5 04
```

And using the update pattern data command will produce the following

```
C-5 10 ~~ .~~
E-5 10 ~~ .~~
E-5 10 ~~ .~~
E-5 10 ~~ .~~
D-5 10 ~~ .~~
F-5 10 ~~ .~~
C-5 10 ~~ .~~
F-5 10 ~~ .~~
C-5 10 ~~ .~~
```

2.6 Song Variables

Most of the options on this screen should be quite obvious, but they are explained here for clarity.

Song Name - should be obvious. Shove whatever you like in here :)

Inital tempo - The tempo that the piece starts with. The tempo calculation is explained under the set tempo command (Txx)

Inital speed - The speed that the song starts with. The speed calculation is explained under the set speed command (Axx)

Global volume - This scales all the volumes in the song. It may be necessary to change this value if there is overload occuring (in the form of crackles in the music) - especially on the Gravis UltraSound, as it is a non-mixing device, and not scaled by the mixing volume.

Mixing volume - This value affects mixing sound devices (ie. all Sound Blaster cards + PC Speaker).

Separation - The separation determines how far apart the left/right panning sounds. It is suggested that for a stereo system (where the speakers are relatively close together) that the separation be around its maximum (ie. 128). For a stereo system where the speakers are placed a large distance apart, a value of 80 may suit better. A suggested separation for headphones is around 40. This value has no significance unless stereo playback is selected.

Old Effects - When Impulse Tracker was first written, some effects were interpreted differently from other formats, most notably vibrato. When you turn this ON, then it effects will be interpreted how the used to be in ST3/MMEdit/... but when it's off, it'll operate how it use to in previous versions.

Differences:
The Vibrato (and Tremelo) used in IT is smoother than how it was implemented in MOD/S3M/etc. It is updated EVERY frame and hence is independant of song speed, whereas the standard vibrato WAS dependant upon song speed. Vibrato is two times 'deeper' with Old Effects on.

Sample offset commands past the end of a sample in IT were ignored, whereas with Old Effects on, the sample is played from it's end point.

Control - This is the option that lets you use Instruments!

When Control Sample is selected, then all the instrument information is ignored. The query to initialise instruments, if accepted, will copy all the sample names to the instruments (if the samples exist) and set up the Note Translation Table for each instrument to point to a sample.

Playback - Mono: When you select mono playback, all stereo commands are essentially ignored. If you are using a SBPro, selecting Mono provides a higher quality output than selecting Stereo and using a central pan.

Stereo: This is the option to go for!
Note that stereo mode requires more processing power than mono (unless you are using a hardware driver)

Pitch Slides - Amiga: This is the mode to choose for compatibility with S3Ms, MODs, MTMs, XMs... when you select amiga pitch slides, you'll often have to experiment to find the adequate slide value.

- Linear: With the linear slides, a certain slide value will always cause the same music "interval" change. An example of what this means is if you slide up a note with speed 8 (ie. F08) at speed 5, then the result will be that the note is raised a tone. Everytime. So a C-4 will rise to a D-4, C-5 will rise to a D-5. (It's not like this with amiga slides! A C-4 *MAY* rise to a D-4, but if it does then the C-5 will rise to a note around E-5!!)

The benefit of linear slides is that if you slide a pair of notes at the same speed, they will maintain the same interval (pitch difference) throughout. Also, if you modulate (transpose) a pattern, you will not have to modify any slide values.

Directories - this shows the current song/sample/instrument directories. If you want to save the current directories as the default to have on bootup, then press enter on the Save button. Note that this save function also saves the palette, keyboard type, info page layout and pattern editing preferences to IT.CFG

2.7 Scream Tracker 3 Users

This section is just for users of Scream Tracker 3 who would like to know the differences between ST3 and IT without working through the whole document. In a nutshell:

1) Differences in use.

a) The Order List, Panning and Variables used to be on F1. In IT, F1 has been reserved for the Help Screen, F11 for the order list and panning, F12 for the variables.
If you REALLY do find this too inconvenient, use the command line switch -k to swap the interpretation of the keys F1 and F11... no text will be updated tho (and it does seem sorta weird!)

b) Samples now have associated Vibrato information and a global volume. The global volume affects all instances of the sample throughout the song. This is useful when you replace a sample, which is at a different volume from the original - you can just adjust the global volumes of the samples to suit.

d) Block functions are no longer restricted to one column. This also means that when you copy entire patterns, you will have to go to the top left of the pattern. Also, you will have to set the number of rows in the destination pattern to the appropriate number if it is different from the source.
This difference may take some getting used to - you may often find yourself doing several block functions in a single columns when you could actually have done it across the whole block!

* To mark blocks, you can use Alt-B and Alt-E, or Shift+Movement *

e) In the pattern editor, the following commands have changed:

1) Alt-T : Track View has been upgraded. Alt-T will cycle through all of the possible 5 viewmethods. You can use Ctrl-1 to Ctrl-5 to quick select a view, or Ctrl-0 to remove a view. Press Left-Ctrl+Shift 1->4 to select a 'different way to edit' :)

2) Alt-X : Pressing Alt-X once will slide the effect data value in a block (useful for controlling global volume, panning or sample offsets in some cases). Pressing Alt-X twice will cause all of the effect data to be

deleted as in Scream Tracker 3

3) Alt-K : Pressing Alt-K once will slide the volumes. Pressing Alt-K TWICE will remove all volume controls in the block. (Works like Alt-X, but on the volume column)
4) Alt-I : Cycle Template control.
5) Alt-J : Volume amplifier. Select an amplification between 0 and 200%.
6) Alt-H : Now toggles view divisions on/off. Makes it possible to view/edit up to 36 channels in the pattern editor!!!
7) Alt-Z : Originally the Zap command (if pressed twice)... now it is effectively a "Block Cut" command. It will copy the block into the clipboard, and then wipe the block clean.
8) Alt-V : Will set all the volumes in the block to the default.
9) Alt-W : Will remove ALL volumes not attached to a note/instrument Therefore, the old Alt-V command can be simulated using Alt-V then Alt-W. The reason for doing this was so that the Alt-K, Alt-W combination would slide all the volumes with notes/instruments associated.

f) There is no chord edit feature in IT. I considered the chord edit feature of ST3 useless and a bad influence on new trackers :)
g) Patterns can be from 32 to 200 rows! (Under Pattern edit config on F2, or Ctrl-F2)
h) Pressing spacebar in the pattern editor will cause the previously entered note/instrument/volume/efffect/effectvalue to be entered. Quite convenient. Hopefully.
i) Alt-F10 now solos a channel instead of toggling all of the channels on/off. On the InfoPage, you can use 's' to solo channels, and 'q' to toggle channels. I felt that this was more usable than the original.
j) The infopage itself has become upgraded... press PgUp/PgDn to cycle through view methods, "Insert" to add a window, "Delete" to remove the currently highlighted window, Tab to move between windows, Alt-Up/Down to move the base of the window up/down.
k) To increase the sample frequency by an octave in the sample list, press Alt-Grey '+' or Alt-Grey '-' to decrease the frequency by an octave. To change it by semi-tones, use Ctrl-Grey '+' and Ctrl-Grey '-'
l) The Sample library is accesible from all screens in the program by pressing Ctrl-F3. The Instrument library is accesible on Ctrl-F4.
m) Each channel has an associated volume. Echoes can be created by just copying one channel onto another, then setting the "channel volume" (Command Mxx)
n) When channels are muted, all commands are still interpreted, but the notes aren't played. What this means is that commands such as Axx, Bxx, Cxx, SBx, Vxx, Txx, Mxx, Nxx will be processed even if they are in a muted channel.
o) Pressing Ctrl-Grey Plus and Ctrl-Grey Minus in the pattern editor will go to the next/previous pattern according to the order list!
p) Samples have sustain loops + Ping Pong loops supported
q) Note off command & Note cut command implemented... Note cut is exactly like ST3's ^^^ (and appears like it too!) Note off appears as "~~~" and releases sustain points.
r) You can enter panning values (between 0 and 64) in the volume column. Do this by pressing '`' to toggle between entry of panning/volume values.
s) You can use samples >64k and 16 bit samples!!!
t) When you use a skip value of 0, the cursor will move across to the next channel. This is a very convenient way to enter chords, especially when combined with the multi-channel selection (whack Alt-N twice)

2) Differences in interpretation of notedata.

a) Global volume changes will affect ALL of the notes playing. In ST3, global volume changes only affected new notes. The range for the global volume command is from 0 to 80h
b) If you specify an instrument without a note, then in ST3, the volume is set to the default volume. In IT, the volume will be set to the default volume *IF* the instrument specified is the SAME as the currently playing instrument. Otherwise, the new sample will be played from it's beginning.
c) Command X (Set panning, not REALLY implemented in ST3) now has range from 0 to 0FFh.
d) Command C (Break to row) now works in HEX. This is so that you can jump to any row in the (next) pattern.
e) New commands:

M:	Set channel volume
N:	Slide channel volume
P:	Slide panning
S5x:	Set panbrello waveform and reset panbrello position

S6x: Pattern delay for x frames.
S7x: Instrument related controls (8 different controls here)
S91: Surround sound on!
SAy: Set high offset
T0x: Tempo slide down
T1x: Tempo slide up.
W: Global volume slide.
X: Set pan position.
Yxy: Panbrello

f) If you use the portamento to command (Gxx) to a different sample in ST3, the sample offset is undetermined. In Impulse Tracker, the sample will be played from it's start.

g) The commands Hxx and Uxx will cause a vibrato half the depth of ST3 if "Old Effects" option is OFF and the vibrato in IT is also smoother than the vibrato in ST3.

3) The greatest difference is the implementation of instruments. Read section 2.5 for more information.

4) Saving S3M modules.

The following guidelines have to be followed to compose 100% compatible S3M modules.

General Stuff
1) *Pitch slide mode MUST be amiga.
2) *All initial channel volumes MUST be 64
3) *No instruments functions can be used. If you ARE in instrument mode, however, Impulse Tracker will translate the pattern data according to the Note Translation Tables.
4) The song message is *NOT* saved in S3M format.

Pattern stuff
1) *The total number of patterns MUST NOT exceed 100.
2) *The number of rows in each pattern MUST be 64
3) *Data is only translated within the first 16 channels.
4) *All notes must lie within the range of C-1 to B-8
5) Several commands will not be interpreted by a standard S3M player (viz. Mxx, Nxx, Pxx, S5x, S6x, S7x, S91, SAy, Wxx, Yxx).
Also, you should have Old Effects (on the variable list) put to *ON* if you want to write S3M files.
6) *Panning controls in the volume column are NOT stored.

Sample stuff
1) *No sustain or ping pong loops can be used
2) *No sample vibrato can be used
3) *The sample global volume must be 64.

4) The Scream Tracker 3 module format supports 16 bit samples and samples >64k although the actual program did NOT. Note that *MOST* players probably ignore these fields and it is likely that the module will NOT be played correctly by a standard S3M player if it contains samples >64k and/or 16 bit samples.

All conditions with an asterix preceding them are checked at save time.

Note that both note off (~~~) and note cut (^^^) will get translated to a note cut (^^^) in S3M format.

2.8 Gravis UltraSound / Interwave / AWE32 users.

For hardware mixed devices (eg. Gravis UltraSound, Interwave and AWE32), there are some limitations that are outlined here.

I've already mentioned a few things throughout the DOC about hardware mixing, but I'll reiterate them here with a few more points.

1) The maximum number of notes that you CAN play with a GUS and Interwave is 32. The maximum number of the AWE32 is *30*.

This was incorporated by effectively limiting the number of active channels to 32/30. So if you play a 64 channel song which is using control:Samples, then you will LOSE the last 32/34 channels. If you are playing a 64 channel song with control:Instruments, you shouldn't miss out on too much. This shouldn't really be a problem in 99.9% of the cases.

2) The GUS playback quality is dependent on the maximum limit of active channels. The GUS can be initialised to play anywhere between 14 and 32 channels. Values above 32 are reduced to 32, and values

below 14 are increased to 14. The lower the number of channels, the higher the playback quality. (14 channels will give 44100 Hz, with 16-bit interpolation - very, very nice quality!). The default GUS driver dynamically chooses a particular number of channels. This doesn't ALWAYS work with all cards, so if you don't like the transition of sample quality as the channels increase, or your GUS seems to stop notes unexpectedly, use the other driver(s).

3) The GUS/Interwave/AWE32 routines occupy less memory than the mixing routines. You will have about 30k-100k more of memory to play around with.

4) The memory allocation routines are very simple and not very versatile. When you are working through the sample library, samples are NOT always deallocated when you move on to the next sample! If you get the message "Out of soundcard RAM", then it may be necessary to press Ctrl-G (or select Reload Samples from the samples menu) to reorganise the samples in memory. Sorry about the inconvenience.

5) Changing the mixing volume does NOT affect these cards.
It may be necessary to reduce the global volume to remove crackle if the output is overloading.

6) The surround sound option will cause the sample to be played with a central panning. This is because I don't know how to get hardware surround without wasting a lot of memory and channels.

7) The panning positions are reduced to a scale of 0->15 in the GUS, so fine alterations in the panning may go unnoticed. The Interwave and AWE32 permit 256 pan positions

3. Before you write to me. (Important notes)

I would very much like to get feedback on this program, but I am already aware of several problems (?).

1) If you're going to write because the program looks 'too' much like Scream Tracker 3, then don't :) I've used Scream Tracker. I loved using it - it was the best, easiest, most convenient program ever, so why not adopt it's design?????

2) The program could easily crash if you try to load a corrupted module. If you've got some modules that you know are corrupted, try to resist the temptation of testing the stability of the program.... in many cases it'll die :)

3) A couple of people have asked about ASCII characters > 128. I'm sorry - I can't let you have 'em on anywhere else except the message editor. The reason is because I've used a lot of the characters for 'graphics' - real time character generation. (the ASCII characters > 128 don't really exist anymore)

4) If you 'overload' your CPU... the subsequent playback can also become distorted! To fix this, reinitialise your sound driver with Ctrl-I.

5) The help screens have only been defined for the following screens:
   a) Pattern Editor
   b) Sample List
   c) Instrument List
   d) Info Page
   e) Order list and panning
   f) Order list and channel volumes
   g) Message editor

   I didn't think it necessary to include help screens for the other screens, but if you feel strongly about including help on certain points/screens, then I suppose I could make the effort...

6) If you have a bug to report, please check through BUGS.TXT first to see if it's listed, and also check through this document - the problem may be in the understanding of how the program works! Otherwise, contact me, AND LET ME KNOW WHAT VERSION YOU ARE USING!

7) Any further GUS click removal is very difficult to achieve. ST3's GUS click removal requires double the number of channels on the GUS to be used and I can't allocate any more! Try out the alternative GUS drivers first

8) Got suggestions for IT? At the moment, I'm really sorry - I barely have enough spare time myself. I can't promise to add any suggestions you make, but please send them along anyway.

You can contact me by writing to:

Jeffrey Lim
9 Wilgena Avenue
Myrtle Bank
South Australia 5064

EMail account:
pulse@cyburbia.net.au

WARNING!!!

If you ask me anything which is obviously contained within this document or the FAQ, I will reply with RTFM (Read the #$%@#$% Manual). Sorry, but I hate idiots who can't be bothered to take their time to have a quick look through this document and write to me with stupid messages which have recently become a chore to answer (You'd understand if you received 4000+ EMails...)

BUT!

If there *is* something that you HAVE made an effort to understand/look for, then don't hesitate to write to me. The worst that can happen is that I reply with RTFM :)

********ALSO********

If you have an incorrect reply-address, don't expect to receive anything.
If you are reporting a bug, let me know what version you are using.

If you find any major problem with the tracker, please try to find the circumstances which cause it to trigger - if you can't, don't worry - write to me anyway. It's just that it's much easier if I know a likely cause of the error. (Don't worry if it seems that the cause is very, very remote - I had a problem in my GUS playback routines which originally caused my palette configuration screen to hang - now that's obscure!)

Send money! I don't demand that you send me anything, (and I don't like spoiling a program by shoving in reminder messages), but if you *DO* use the program and think that it's worth contributing a small amount towards, then please do so... it would be GREATLY appreciated. Remember.... even a few dollars will be appreciated! It's very unlikely that I'll make a huge profit (I doubt I could get close to sufficient money from this program to make it even worth $1 for every hour I put in!). I won't be sending out any nice thankyou notes, or bound manuals... but I will include your names in the future versions of the tracker and I will EMail you each new version the minute the are released if you wish.

I believe that software *SHOULD* be cheap - you've spent perhaps a few thousand getting your computer, so you wouldn't want to spend TOO much more on programs! If you think this program is worth $5, then send $5. If you think it's worth $10, then send $10. If you think it's worth $1,000,000,000... don't let me stop you :)

So... get together with a couple of friends and send me $10... If you send me a significant amount, I'll put together a "special contributor's" list.. (Oeeerrrrr :))

Methods of payment:
  International Money Orders - (NOT postal orders though!)
  Cash (Make sure it's securely wrapped!) - it doesn't have to be Australian currency, although it'd be nice if it could be!
  Bank Transfer - EMail me for my details

4. Closing Words

I would like to thank the following people:

Psi: For introducing me to tracking with the brilliant Scream Tracker 3

Purple Motion: For being my music hero.... I still think "When the Heaven's Fall" is *THE* best S3M ever written.

Red Haze: For constantly bugging me to get the tracker done, and for his effort in testing out the program and making suggestions. (He's directly responsible for the inclusion of Ping Pong loops - I originally couldn't be bothered :))
Also extensive testing of later versions of IT...

The WALKER: For his flow of ideas & beta testing and getting the PoP DiskMag finished.... tomorrow. Or is that next week?
OK... so he has done it. Finally.... 6 months late. :)

Perception: For his constant support over the many years he ran his BBS

Psibelius: For being the nice friendly Epinicion founder that he is.

Dominic: For his thoroughness in beta testing the program, and providing suggestions. (He came up with more than double the number of suggestions than almost anyone else... even though many of them were not finally implemented :))

Chris Jarvis: For his effort in beta testing the tracker and producing
Firestorm (released with v1.00 of the tracker)
Pale Dreams! (released with v1.01 of the tracker)
Firepower (released with v1.03 of the tracker)
Sidewalk (released with v1.05 of the tracker) and
Fallen World (released with v2.01 of the tracker)
These are ALL amazing pieces of work! (Yes, I'm jealous!)

Zilym Limms: For his technical help with the tracker and his love of Fast Tracker 2. Hahaha. :)
Also for his information and source code for the PAS 16.

Benjamin Bruheim: For his super list of contributions to v1.01

ZaStaR: For his effort in putting together ITF - the font customiser! and ITTXT - the text file importer for IT's messages!
Also for being the willing target of a barrage of beta versions, a great contributor of ideas, a musician with whom I could exchange work and for being a great dude all round!

Emmanuel Giasson: For creating the utilities MMCMP/MMUNCMP/MMTSR - the music module compressor!

Diablo: For creating the Impulse Tracker directories on ftp.cdrom.com!

Siren: For providing me a contact to obtain an Interwave card!

AMD: For providing me with an Interwave card to program for!

To the other beta testers:

Eric Bonython
Alistair Watts
Alex Bates
Delta X
Andy Chen
Clef
KXMode
Phorte
Julian Ellis
David Rohrsheim
GD (Grave Digger)
ShawnM
Emmanuel Giasson

And anybody there who uses this program!!!

Big thanks and thumbs up to Advanced Gravis for releasing their Software Developer's Kit (SDK) free of charge!

Congratulations to Creative Labs for finally releasing their SDKs free of charge!

Especially *BIG* thanks to AMD for providing me with an Interwave card to use and the SDK!

Finally... I have a request.....
If *YOU* write something with this program... share it with the rest of the world!!!

If you have an iNet account, you can upload your songs to:
ftp.cdrom.com/pub/demos/incoming/music/songs/it
- remember to ZIP up your song, to use a *lower* case filename and to also upload a short description of your song in a .txt file, otherwise your file will just be deleted.

If you know of any other places where modules can be uploaded to, let me know and I'll include them in the above list.

5. How to get the latest version of Impulse Tracker

There are several places around the internet where you can find the latest version of Impulse Tracker:

```
     1) ftp.cdrom.com/pub/demos/incoming/music/programs } Check (1) first!
     2) ftp.cdrom.com/pub/demos/music/programs/trackers }

     3) http://www.noisemusic.org/it
                - American Site
     4) http://www.maz-sound.com
                - Music and Tracking Site
     5) http://www.unidev.com/~logic/music/it
                - IT Resource Central
     6) http://www.musica.org/impulse
                - Spanish Site
     7) http://www.mixbbs.demon.co.uk
                - UK Site
```

If you want to have the latest version sent to you via EMail, or at least be notified of new releases, then you'll have to send $omething to me :) Check section 3 on how to contact me.

6. Legal stuff

No matter what happens, no matter how bad, I'm not going to be held responsible.

That's basically the same as any other license agreement, except you should be able to understand this one without an interpreter :)

YOU MAY NOT CHARGE ANYTHING FOR THIS PROGRAM - DISTRIBUTORS WHO ARE INTERESTED IN THIS PROGRAM MUST WRITE TO ME FIRST AND HAVE MY APPROVAL! (yeah, even by snail mail if you don't have EMail access).

USE OF THIS PROGRAM COMMERCIALLY IS EXPRESSLY FORBIDDEN WITHOUT WRITTEN AND SIGNED APPROVAL FROM ME. COMMERCIAL USE INVOLVES ANY USE OF IMPULSE TRACKER IN WHICH MONEY IS INVOLVED (SPECIFICALLY THE CREATION OF ANY MUSIC WHERE PAYMENT IS INVOLVED). THIS PROGRAM IS ONLY FREEWARE FOR NON-COMMERCIAL USE.

Impulse Tracker and Resonant Filters

Wanna know how to get resonant filters working in IT? Read on.

Implementation

So far, resonant filters have only been coded into the MMX drivers - so any soundcard which has an MMX driver for IT will support resonant filters. Of course, this means that your computer has to have MMX before you can run them. To hear resonant filtering, you'll first need to select "Filtered" mixing on Shift-F5.

Do NOT write to me about non MMX resonant filtering.

Resonant filters CANNOT be included with hardware GUS / Interwave drivers. The reason is because these chips do not support resonant filtering in their mixing algorithms.

The AWE 32 driver has *approximate* support to IT's software resonant filtering. Songs written using resonant filters on the AWE32 will not sound exactly the same with other drivers.

The diskwriter has all resonant filtering code, of course. (No MMX required)

Note that the only external player to have resonant filtering implemented is MikIT. If you use any other player to play your songs that use resonant filtering (including earlier versions of IT), they will not be played correctly.

First note

If you do use filtering in your songs, you probably should embed your MIDI Output configuration into the .IT file. This makes the file slightly bigger, but it ensures that your song will be played correctly on any filter-capable driver on any computer. This is selected by turning the "Embed MIDI Data" on the MIDI screen (Shift-F1) to "on".

Simple filters

For most users, this is all that you will need to know.

The default configuration for IT (copy ITMIDI.CFG to your IT directory) will recognise Z00->Z7F as set filter cutoff frequency and Z80->Z8F as set filter resonance.

```
 Z00 is the lowest filter cutoff, Z7F is the highest filter cutoff
 Z80 is the least resonance, Z8F is the highest resonance
```

If you wish to reconfigure the resonant filters or perhaps create some extra shortcuts, then read below!

How the drivers recognise filters

The drivers know what to filter by intercepting MIDI messages. This does NOT mean that filters will require any sort of MIDI equipment, just that the mechanism to instruct the driver to filter a particular note within IT itself is made via the MIDI interface.

The instructions that the drivers understand so far are:

```
  F0 F0 00 <value> - Set filter cutoff frequency to <value>
  F0 F0 01 <value> - Set Q factor (resonance) of filter to <value>
```

In each of these cases, <value> is between 00 and 7Fh. Values above 7Fh are ignored. Note that if filter cutoff is set to 7F and Q is set to 0, then no filters are applied.

How to tell the drivers these Instructions

OK.. so how can we tell the drivers these instructions?

For a full explanation, check MIDI.TXT - a short explanation is provided here.

First of all, go to the MIDI Output configuration screen in IT. Do this by pressing Shift-F1, then clicking on the "MIDI Output Configuration" button.

You will see several MIDI configurations, then SF0->SFF then Z80-ZFF (that bottom window is scrollable).

Using Z80 to ZFF

Z80->ZFF are the easiest to explain.. so I'll explain them first.

If you type in "F0 F0 01 3F" next to Z80 (make sure that you have the letters in upper case), then whenever you use Z80 in a pattern, "F0 F0 01 3F" will be sent to the driver. If you refer back to what instructions the driver understands, you'll see that this means "Set filter resonance to 3F".

A few more examples:

```
 Z81 = F0 F0 00 40 - set filter cutoff frequency to 40h
 Z82 = F0 F0 01 20 - set filter resonance to 20h
 Z83 = F0 F0 00 10 - set filter cutoff frequency to 10h
```

Using SF0->SFF

SF0->SFF are slightly more difficult to explain.. but hopefully a few examples will make their usage clear.

When you use Z00 to Z7F in a pattern, they do not directly translate in the same way as the Z80->ZFF do. Instead, they set a variable internally called 'z' that gets substituted into one of the SFx commands.

Example 1 - If you set SF0 = F0 F0 00 z (on the MIDI Output configuration)

Then using Z01 will cause "F0 F0 00 01" to be sent.

```
 Z01 = F0 F0 00 01 - Set filter cutoff frequency to 1, as above.
 Z10 = F0 F0 00 10 - Set filter cutoff frequency to 10h
 Z30 = F0 F0 00 30 - Set filter cutoff frequency to 30h
 Z50 = F0 F0 00 50 - Set filter cutoff frequency to 50h
```

Example 2 - If you define:

```
 SF0 = F0 F0 00 z
 SF1 = F0 F0 01 z
```

Then:

```
 SF0               - Set Zxx to use SF0
 Z30 = F0 F0 00 30 - Set filter cutoff frequency to 30h
 Z50 = F0 F0 00 50 - Set filter cutoff frequency to 50h
 SF1               - Set Zxx to use SF1
 Z20 = F0 F0 01 20 - Set filter resonance to 20h
 Z3F = F0 F0 01 3F - Set filter resonance to 3Fh
 Z50 = F0 F0 01 50 - Set filter resonance to 50h
 SF0               - Set Zxx to use SF0
 Z30 = F0 F0 00 30 - Set filter cutoff frequency to 30h
 Z10 = F0 F0 00 10 - Set filter cutoff frequency to 10h
 SF1               - Set Zxx to use SF1
 Z20 = F0 F0 01 20 - Set filter resonance to 20h
 Z30 = F0 F0 01 30 - Set filter resonance to 30h
```

Note that the default startup configuration for each channel is SF0, so the first SF0 in example 2 is unnecessary.

Resetting the Filters

Since the filters are driver related (and IT.EXE really doesn't know about their existance), they are not reset automatically when you stop/play a song. If a MIDI Reset (FFh), MIDI Start (FAh) or MIDI Stop (FCh) message is received, then the driver will reset all of it's internal tables. The default configuration will send both MIDI Reset and MIDI Stop commands.

Final Notes

Umm.. Enjoy :)

- Jeffrey Lim

Hints for Composers
-- Pulse

Well, I'm not the best person to ask for hints, but here are a few anyway.

1) Never release your first song. There are very few people who are gifted enough to really make a quality song the first time - it's all practice and experience! Once you *HAVE* finished a song, listen to it a couple of days after... see whether you can view it from another point of view.

2) For channel echoes, use the Mxx command in a second column - this will save you from adjusting volume related effects (ie. you can leave all the Dxx commands alone, and it'll sound right)

3) Don't be afraid to create multiple instruments from the same sample! The reason for why I created instruments the way I did was so that you could have different *articulations* of the same sample. You can achieve this by playing around with the envelopes, fadeout, NNA - whatever.

4) Listen to other tracked music. Try and learn how other composers have achieved the sound they did. Experiment yourself.

5) Start by writing music that *YOU* really like listening to - don't try and write am orchestral piece if you don't listen to it - it'll show.

6) Take the time to tune all your samples as accurately as possible! To do this, play a long, clear, looped sample, then move to another channel (using '.') and tune ALL your other samples to this one sample (so they all have the same reference). Many potentially excellent modules have been spoilt because they were poorly tuned. Of course, this doesn't count the cases where samples are intentionally slightly sharp or flat for effect (which should be the rarity instead of a rule).

7) Try to avoid having too many samples at central panning - if you modify the initial panning - you should be able to 'fill' out the sound with very little extra effort. Or perhaps if you use instruments, you may want to play around with instrument's default panning...
Pitch pan separation also provides a very convenient way to achieve a nice pan.

8) To find the 'perfect' loop:

   a) If you have a GUS/IW, first turn the loop off, then reload all GUS samples (so that their entire waveform is loaded).
   b) Now, select either a forwards or ping pong loop. Only select forwards if you have a sample which has the same amplitude at both ends. If you have a sample which has vibrato incorporated into the sample, then you'll probably find ping pong loops inappropriate. If the sample has an obvious reoccuring shape to it's waveform, try to account for that when you select your initial guess at a loop.
   c) Play a note at a MUCH higher pitch than you'd normally play it at. Then, hold down '+' (or '-') on on of the loop boundaries to find a region of lowest clicking. Then adjust it carefully (one byte at a time) until you find the best loop location. You will normally need to change both beginning and end points of a ping pong loop to find a nice loop, whereas forwards loops usually only require either loop end or beginning to be modified.
   d) Now that you have a decent loop at this pitch, decrease the pitch (typically by an octave)
   e) Repeat steps (c) and (d) until you have a nice loop at the pitch that that sample is played at.
   f) Once you've finished and if you're using a GUS, press Ctrl-G (to reload the Gravis' samples) and do a final check that you have an appropriate loop.

   This method works very well MOST of the time - don't forget that the '+' and '-' keys can be used to easily modify the loop - and the changed loop is taken into account when you change it (ie. you don't need to replay the sample).

9) If you want to make a song realistic, try to imagine how the instrument would be played. Pretend you are a musician when you write a part..
Also, if you use an instrument such as a piano, try to use more than a single piano note - a real piano will ALWAYS have more than one note playing at a time - use some chords, etc.

10) For a nice fill to the sound, try to balance the usage of low and high frequencies. Songs with too much bass and too little treble sound rough, songs with too much treble and too little bass sound insubstantial.

Hints for New Composers

-- John Hawksley (a.k.a. Greebo)

1) Listen!

2) Spend a day figuring out every feature of the tracker. Yes, I'm talking about all the effects and all the keys. ST3 is widely acknowledged to be a bitch to learn, but is (sorry, *was*) the most powerful tracker out. Once you have all the keys and functions sorted, you'll be ripping around IT's in no time. You can leave the advanced instrument stuff for now.

3) Listen to other tracks, find out how the nice-sounding bits are done. (ie look at the effects and volume/pan column).

4) Be different. A lot of .MODs are in the same style. Sure, if you like this and feel comfortable with it, then go for it! But if you want to create a new feel -- do that too. People are always ready to try new styles. I personaly enjoy arranging (that covers a lot of styles) but you might like composing rock tracks, for instance. So do it!

5) Samples. Be selective. Sort all your samples into directories. If you have an editor, the trim thein sample; try to remove the noise or click at the start. Remember -- samples are the building blocks from which we craft music. If the samples are bad, the music will be too.

6) Tune the samples! When you rip a sample or create one yourself try to do it at the same pitch, or tune it (using the speed value) so that everything is uniform. This will save much hair-pulling later as you try to figure out why half the piece seems to be in G# major and half is in Dflat minor.

6) Chords. Originaly, people used to sample whole chords to save sample space. Now we've got this wonderful IT with it's gazillions of channels. From ST3 onwards, I have been contructing chords from notes because I had the space to do so. The sound is better and is more of a professional approach. However (there's always a 'but'): be very careful! If you decide to construct a chord rather than use a single sample, some musicianship is required. Simple major chords are easy, but inversions really add to a piece. If you are able to do it this way (look at some piano parts to any of my stuff, for instance), you'll get s professional, crafted sound. But it does take a long time before you'll get a smooth flowing part.

7) Saving. Okay, so IT hasn't crashed on me yet, but when (if) it does, I'm not going to loose an hours work. Save regularly. Never use IT or ST3 under the GUI in 95 and under Windows 3.1; I found that occaisionaly, windows would do some swapping while ST3 was saving and the module would be corrupt; but ST3 said it was saved ok. Lesson learnt.

8) Releasing. FTP sites are hard to come by these days. Probably the best method of release is to uuencode your work and post it to alt.binaries.sounds.mods newsgroup.

Hope these are of some help. Remember to visit the Mod Resource Web at http://www.armory.com/~greebo/mod.html

I can be contacted at greebo@armory.com.

Good luck!

John H.

Hints for Composers

-- ToalNkor / Realtech

TIP FOR LOADING EITHER LEFT OR RIGHT CHANNEL OF A STEREO SAMPLE :

Load the sample as usual and then follow these steps :

If you want the LEFT channel : Just divide the length by 2 by using Ctrl-F. This will delete one byte out of two, and therefore only the "first" sample (the left one) will remain !

If you want the RIGHT channel : Cut the first and last byte of the sample (By looping it and using Ctrl-B and Ctrl-L). If the original sample sise was X, then the actual size should be X-2. From now on, just follow the same indications as for the left channel and tadaa... your Right channel sample is ready for use !

After all these operations, dont't forget to multiply the mixfrequency by two to get the original samplingfrequency back !

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Hints for New Composers
-- StereoMan

1) The easiest way to produce flanging like effect is to play same sample in two channels (they must have exactly the same pan-position) and lower or higher the playing frequency of one of the samples - ie:

```
          1            2                             2          (1 is same)
   ~... .. .. Xpp~... .. .. Xpp~              ~... .. .. Xpp~
   ~xxx ii xx ...~xxx ii xx EE1~      or      ~xxx ii xx u11~
   ~... .. .. ...~... .. .. ...~              ~... .. .. u00~
                                                         u00 and so on.
   <pp>  has the same value in the two channels.
   <ii>  is your instrument number.
   <xxx> is the note you play the sample in.
```

2) You can use the above mentioned effect, but instead of having the channels with the same pan position you can put them as Left and Right (full) ie:

```
   ~... .. .. X00~... .. .. XFF¦
               . . .
```

this will give you a smooth three dimensional sound.

Note: This effect has not been tested on SurrounD equipment - the results are li'l unpredictable.

3) Quite a good way to make reverb-like-echos is shown below:

Let's say You have some sequence playing in one channel. Put the same into another channel and insert one or two (or more) rows before the beginning. Now set all volumes to zero (alt-v) and clear volumes which are not associated with notes (alt-w). Then apply a Dx0 effect (x=1..4 or more) for example:

```
    ~n1. i. .. ...~... .. .. ...~          The results are very good.
    ~n2. i. .. ...~n1. i. 00 D20~          Once you get used to this you can
    ~... .. .. ...~n2. i. 00 D..~          achieve !very! smooth sound.
    ~n3. i. .. ...~... .. .. D..~
    ~... .. .. ...~n3. i. 00 D..~          The samples must not be too short
    ~n4  i. .. ...~... .. .. D..~          so Dx0 can take effect.
    ~... .. .. ...~n4. i. 00 D..~
```

4) If you make the above channels with different pan positions (x22 and xDD) or (x80, s91) - the results are stunning :)

5) Take your time to read the whole help (yes, the whole of it) - you'll be surprised to find what hides under your keyboard :)

6) Make your tunes as small as possible. People are not quite happy to find they have a 3 or 4 Megs of crap on their already full HD drives. Remember: the smaller = the easiest to spread.

7) NEVER start tracking if you're not into the right mood to track. You'll only loose time and perhaps make another crappy tune.

8) Funny, but I've found that making your own color scheme truly inspires!

9) Experiment! Play around with the effects, envelopes and NNAs. They all make music sound more realistic!

George Marinov a.k.a. StereoMan - <georgehm@bse.bg>

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Hints for composers
-- Ilpo Karkkainen

- If you listen only one kind of music, it will shut your mind from others. Be versatile. When you listen lots of different kinds of music styles, it also makes your composing a lot more wider and colorful.

- When listening to music generally, try to sometimes consentrate to something specific, for example backing vocals or drums. It helps you realize the whole. It's also good to try listen what different notes there are in a chord that you hear. At least to me, it has been very helful in chord progression.

- Details make the whole. Use them wisely, though. Too much details make the song sound bad. I've noticed that in some of my songs.

~~~~~~

Onix4MAN's hints

1) CLEANING A WAV FILE UNDER IT.
2) CREATING NEW SAMPLES WITH IT.
3) 3 (4?) METHODS TO MAKE YOUR MODS SOUND MORE SPACIAL..

1) CLEANING A BAD SAMPLE UNDER IT:

To clean up samples that click at their start (or end) because the waveform has an error at its start (still or end), without going under a wav-editor:

```
-turn on Loop
-start the loop at 100 bytes for samples > 10000 kb
                    50 bytes for samples < 10000 kb
-then do 'ALT-B': Pre-Loop Cut Sample
-then turn off the loop
-do the same at the end of the sample with 'ALT-L' if the wav clicks
 at its end
```

The numbers of bytes given is OK for often met clicks, if your sample is really bad, just increase it... ;)

2) CREATING NEW SAMPLES WITH IT:

-You simply have to edit one pattern composed of several samples. (eg. Compose a Break-Beat on that pattern)
-Put this pattern at order 000.
-Save this module.
-Restart IT in Disk-Writer mode.
-Load your module.
-Play it: it is now being written as a wav file on your disk.
-Restart IT normally.
-Load that new sample and use hint 1) if it has a blank at its end to shorten it.

3) 3 (4?) METHODS TO MAKE YOUR MODS SOUND MORE SPACIAL..:

-Let's start with the 4th method: it's the Surround.. :)
But if your card can't afford surround.. Use one of the 3 following methods:

These methods are in fact three times the same but with 3 different way.
I'm sure you knew at least the first (and probably the 2nd too) ;)
These 3 methods require 2 channels.

For the 2 firsts, you have to set the panning of the Sample/Instrument somewhere (in 'Order list and panning' or on the Sample List [F3], but you'll have to load twice the sample, or on the Instrument List [F4], or on the Pattern Editor itself [F2], but you busy the volume column or the command column..)
In the following examples, I've set the Panning in the Volume column (press the key below Escape to do this)

```
a)    row        CHANNEL 1         CHANNEL 2
      000      C-5 01 00 .00     ... .. .. .00
      001      ... .. .. .00     C-5 01 64 .00
```
~~~~~~

```
        002       ... .. .. .00      ... .. .. .00
```

```
   b)   row         CHANNEL 1         CHANNEL 2
        000       C-5 01 00 .00      C-5 01 64 SDx
        001       ... .. .. .00      ... .. .. .00
```

With 'x < Speed Value' This second method is more precise!
You can even write SD0 (ie. 0 as x)

c) The last method is the more interesting if you knew the others, because it does waste your volume column neither the command column! So they remain free for other effects! :)

* This time, you have to be controlled by Instruments (F12 to select this).
* Then you will need exactly the too same instruments:
  - On F4 Screen, select a blank lign and type 'Alt-P'
  - Type the lign where your instrument is... Validate!..
* Then push the panning button:
  -set the pan to 00 for your first instrument
  -set the pan to 64 for your second instrument
* Then FOR ONLY ONE of those 2 instruments:
Press the Pitch Button and go to edit the envelop:

```
    -First node: tick  00 ;)
                 value 00

    -Second node: tick  01
                  value 'Whatever_you_want', (-)1 or (-)2 suggested

    -Last (3rd) node: tick  02
                      value 00
```

Doing this, you've created a delay between your 2 instruments.
To end, place them on the same row on the pattern editor (F2):

```
        row         CHANNEL 1         CHANNEL 2
        000       C-5 01 .. .00      C-5 02 .. .00
        001       ... .. .. .00      ... .. .. .00
```

Notes:
======
*) You don't have to set the pan to its maximum (00 and 64/FF). You had better do it for one of your smp/inst. And then, for another choose 16 and 48 (decimal), or...

*) The third method works because we do not hear the pitch change in most cases since it is quite quick, but I suggest you do not use this method for a piano because it's an example where you'll hear the pitch change and it will sound very ugly: BAAaah! ;)
But it work with Violin and many others.
It may also depend on the speed of your song (time between ticks)..

- Nicolas ARROUET (Onix4MAN) o4m@mail.cpod.fr

~~~~~~

Hints for new composers.
- Nacho Segura

About quality of sound, cognitive science, a more convenient composing, sound experiments and degrees of freedom.

1) Some composers (trackers) recommend to work with 128 rows and half the speed (the less the faster). It's supposed that this gives you more control over the tracks, but that's not totally true.

- The track doubles its length, so you see the half. You have to move more times and jump more lines every time. Is it important? Register in a sheet (or two) how many times do you jump through the pattern.

- I've examined several songs that use this technique, and this is my conclusion: THEY DON'T NEED IT!!! Even lines are empty or have effects.

The small amount of control gained doesn't compensate the ergonomical problems. The easiest the best, less interferences between you and the music. Rookies could think that it only makes you be slower. This is a problem, but it's not THE PROBLEM. When you forget twelve times what the hell did you write in top of the pattern and in which track you'll understand...
~~~~~~

2) Work. Lots of trackers are proud to say that they are very fast. That's not a virtue, it means less work, less variety, a shorter melody, much less chords, no harmony, sounds not perfectly adjusted, and the most important thing: Repetitions until the Eternity. You haved lasted two months writing this four minutes long song? Show me what you did, I'm really interested!

3) Discover Scroll-Lock. Load a song, press play, see what happens with the cursor... and press some notes.
Cool!!! Isn't it?

4) Never use 8-bits or low-quality samples if you can avoid it. The quality of a song depends on the quality of sounds. "More memory than expected" is better than "crappier than expected".

5) Analogic synthetised instruments can produce strange interactions. An example: WARMPAD.PAT (a Gravis Patch) sounds really nice, but this chord produces a strange noise that doesn't exists when we play the same notes separately: C-3, D#-3 and F-3. Upper octaves don t provocate this phenomenon.
NOTE: There are several versions of Gravis patches.

6) Use a global volume as high as possible. It not only gives you a better signal-to-noise relation. It also gives to IT more degrees of freedom for volume fades.
Make an experiment: Plug the headphones directly to your soundcard, set the global volume to 5-8 and make a fade out from 64 to 0 (don't use envelopes, make it in the volume comlumn of the pattern). You should listen the volume JUMPING (not sliding, jumping!). In Scream Tracker is even worse.

7) Don't be messy allocating tracks (channels, columns... you know). All the percussion grouped in adjacent tracks, the chord grouped, an empty column (or more) separating every group of instruments, so you can write fastly this new idea appeared two seconds ago, without having to go to "Track 21". It also allows you to write and remix fastly. Everything has its own place and you can disorder and reallocate patterns without knowing if that loop has been cut, or where do I have to put a NoteCut command (^^^) to shutdown the analogic looped bass. It seems more complex when you begin (pattern is wider), but it's much better, easier to use.

8) Print the manual and bind it. And when you have done this, RTFM (you know, READ THE #%&@$# MANUAL!!). You'll be surprised.

9) Make an economic contribution. I think he has worked hardly and Impulse Tracker is the only tracker that gives tracker songs a proffesional sound and accoustic. Don't be apologized for sending eight dollars. Is better than zero. Even if you don't want the ITWAV.DRV you should contribute, at least with a simbolic quantity. He has won it.

Hints for composers
-- Joakim "Acoustic" Back

1) Dont use the same bassline, piano chords or whatever the whole song. Remember that a real drummer wont just sit there like a drummachine, nature will make him tap sometimes and he realy wants to make some fills sometimes.

2) Feel the music. You dont use hammering industrial drums in a soft, smooth gentle song. And dont place a soft panflute in a blasting hardcore song.

3) Use the right volume. Keeping the volume low on an instrument and then suddenly higher creates a feel of power and rush. Use it.

4) There are different ways to make a solo stand out.
   a) high volume. Having a high volume will instantly keep it in the focus. Be aware that to high volume will make it stand out to much and maybe not fit in the picture anymore.

   b) high or low pitch. If you have a lot low and middle note instruments the solo will be clear and bright as high pitched. As said in other hints, keeping the others too low or too high will sound terrible.

5) If you play the piano and have a midi keyboard - use the midi support!

This will make you see that you play the piano with a lot more feeling than when you track a song with the computer-keyboard.

6) Accept failure. Dont get all angry just because your song went totaly nuts. This happends all the time. Your songs will be better and better the more you use IT.

7) Use IT alot! Play around with IT, make crazy songs. This will make you learn ITs features and ways to make nice effects. You wont understand IT by reading the effects from Axx to Zxx, or reading hints like these, but mostly by using the effects and using IT, only training will get you to the top. IT is like a sport, people that dont use it, dont get a thing about it, but when you get the hang of it, its going to flow.

8) Learn IT in steps. Begin with some simple samples and a few patterns, then learn a few simple effects like Exx and Fxx. When you have learned them, go to a new step. Wait with the instruments (F4).

9) Make a keychart. Write down some of the keys on a piece of paper, after a few days you will probably know most of them. When you know almost all you will understand that using only the keyboard is MUCH faster than the mouse.

Thanks for reading, I hope it will help you somehow.
/ Acoustic
n98joab@tycho.helsingborg.se

~~~~~~

Hints for Composers
- Maarten Van Stien
- Crystal Score / The Black Lotus

He! ye like simple solutions? Here's one!

You might have used the diskwriter for simple drumloops! You also might have loaded the wav in programs like Soundforge (the king!) to add nice nice stuff like reverb, eq, dynamics etc...

As you might guess when you write one period containing a drumloop and you add reverb, then the start of the sample doesn't have reverb at all, while the end of the drumloop as tons of reverb. Apart from the fact that it sounds lame in most cases, LOOPING the sample sounds like hell! So what you do is diskwriting the same loop twice or more. Then add reverb in your sample editor and you'll notice that the second period contains the 'reverb' of the first period. This second period can be looped perfectly. As long as you know where to find the looping-points!

Now, for simple drumloops with a little bit of reverb it's dead simple.
But for complex loops with TONS of reverb/delay/crap&more it might be quit difficult!

Solution:
* make an extra .IT with the same BPM/frames as your drumloop .IT
* add a simple, short and immediatly-starting sample at the beginning of each period (in most cases: on pos 000, 016, 032, 048 etc.). These are some sorta metronome instruments I guess..
* diskwrite
* find the first sample of the 'metronome-instruments' in a sample editor.
* add markers at that place.
* Mute/Silence the samples so that your metronome-wav ONLY contains markers
* copy your complex drumloops or whatever-loops in mem. like ctrl-c
* MIX to the metronome wav.

Now you have markers in your complex drumloops! And if you did the above stuff right, you have perfect loops!

Make sure the metronome-samples start immediatly! Otherwise use the offset command (C-4 10 63 010) or something..

ok.. have a nice diskwrite!

Crystal Score/The Black Lotus
Maarten.Vanstrien@student-kmt.hku.nl
~~~~~~

## Impulse Tracker MIDI Supplement

Introduction

The biggest new addition to Impulse Tracker 2.12 has been support for sending data out through the MIDI protocol. Earlier versions of Impulse Tracker (IT) have had support for incoming MIDI data, so it was possible to play on a 'MIDIfied' keyboard and have IT 'track' all the notes down in the pattern in realtime.

MIDI out, however, is a completely different ballgame. You in fact don't have to read this supplement at all if you wish to get MIDI working, but in order to gain full usage of MIDI out, you will have to read this in order to understand how IT's MIDI support fully works

MIDI Support & Soundcards (Modified by Pulse)

At present, the only sound devices IT supports for MIDI output are the AMD InterWave and the Creative Labs Sound Blaster AWE32. Support for other soundcards is via the generic MPU401 driver. To use this driver, you must run "IT /S19 /A<address>" where <address> is the address of your MPU401 compatible card (eg. "IT /S19 /A330"). Note that the generic MPU401 driver does *NOT* support sample playback at all.

MIDI And AWE32 Soundcards (Pulse)

I have found that the Windows '95 drivers for the AWE32 are somewhat buggy. On my computer, I can use MIDI In/Out in Impulse Tracker *IF* I disable the MIDI support within Windows '95. Otherwise, it is simply not recognised. To disable the MIDI support in Windows '95, right click on "My Computer", "Properties", "Device Configuration", "Sound, Video & Game Controllers". Double click on "Creative Labs SB16 or AWE32" and go to the resources. Uncheck the "Use Automatic Settings" box if it is set and change the basic configuration to one that does NOT include the MIDI Ports 300h or 330h. (For example, Basic Configuration 0 has only 220h, 5, 1, 5 on my computer and IT works fine in a DOS Box)

Enabling MIDI

To enable MIDI output, you must first have IT on Instrument control mode rather than Sample control mode. To do this, press F12 to go to the Song Variables screen and switch control to 'Instruments' if you haven't already done so.

MIDI & Instruments

MIDI Channel

New to the Pitch section of the Instrument screen (press F4 and select the "Pitch" button) are the sliders "MIDI Channel", "MIDI Program" and "MIDI Bank". Once the MIDI Channel value for that instrument is set to a value other than zero, IT will send out MIDI data whenever that instrument is encountered in a pattern, on that particular MIDI Channel. What data IT actually sends through the MIDI port will be a note on command, although this will be discussed in more detail later.

MIDI Program/MIDI Bank

The MIDI Program and MIDI Bank sliders work in a similar manner to each other. If they have a value set to -1, IT will not transmit a program change message nor a bank change message for that instrument. If you specifically set a MIDI Program for that instrument, IT will send a 'program change' message along with the 'note on' message.

The MIDI Bank instrument setting is also the same; IT will not send a bank change message if the MIDI Bank slider is set to Off (ie: has a value of -01). If the MIDI Bank setting is active for that instrument, IT will send it along with the note on message as well.

Summary

IT will always send a 'note on' command for a particular MIDI channel whenever a MIDI instrument is encountered in the pattern. (A MIDI instrument is simply an instrument where the MIDI Channel value has been set to something other than "Off"). IT will also send a program change command and/or a bank change command along with the note on command if they are set active.

More Advanced MIDI

How MIDI Works

MIDI is not a file format (like IT is a module format) nor is it even a file layout. MIDI is a computer protocol (or language) which is used to communicate between devices . You may like to think of it as a network, where the MIDI cables are the cables you lay between computers, and MIDI is the network protocol (such as Novell NetWare, Windows Networking or TCP/IP) used to communicate between the sound devices. When an instrument is said to be 'MIDI compliant', that means that it has support for the MIDI protocol and understands MIDI messages.

The .MID file format is simply a way to store these messages. It is a collection of MIDI data, and when a .MID file is run through a MIDI player, all the MIDI player does is send the data in the .MID file out through the computer's MIDI port. In a network analogy, if you can imagine that every single transaction run through the network was being logged to a file on your hard disk; that every single byte was being recorded to a logfile, this is what a .MID file is.

An example of MIDI Communication in IT

You've read above that when IT encounters a MIDI instrument in the patterndata, it sends a 'note on' command, which is defined in the MIDI protocol to be 'Play this note on this MIDI channel at this particular velocity'. (Velocity is similar to volume). If you play a MIDI instrument which is mapped to MIDI channel 2 at C-5 with a velocity of 64, the actual data which IT sends out to the MIDI port resembles something like this (in hex):

```
       Note On with parameters;
              MIDI Channel: 2
              Note: C-5
              Velocity: 64

         Data that IT sends out (hex): 91 3C 40
         Data that IT sends out (decimal): 145 60 64
```

We'll run through each of these three bytes step by step.

The first byte (91 in hex, or 91h) is the actual 'Note on' command. It tells the receiving MIDI device that the data which follows is part of the 'note on' data. An analogy which trackers may find useful is the effect column. There, you have an effect command and effect data; for example, the effect 'G20' can be split up into two parts--'G' and '20'. The 'G' part is the actual effect command which tells IT that you wish to perform a portamento, and the '20' part is the effect data, or in this case the spe ed at which the portamento should occur. In this MIDI example, the '9' is the MIDI command and the '1 3C 40' is the rest of the data for that command.

Now, the second digit (1 in our example) specifies the MIDI channel. MIDI channels are 0-based; that is, if you want to send to MIDI channel 6, IT specifies 05 for the actual data. MIDI channel 10 is 09h, MIDI channel 14 is 0Dh, etc. Here we're sending to MIDI channel 2, so the value sent out over MIDI is 01h.

The second byte (3Ch) is actually the note to send (C-5 here). In MIDI, all command (parameter) data is between a scale of 00h-7Fh (or 0-127 in decimal). Notes are transmitted the same way - via numbers. If you imagine C-1 is sent with a value of 00h, C#1 i s 01h, D-2 is 02h, etc, then the note we want to play, C-5, has a value of 3Ch (60 decimal).

Now the first and second bytes are done with, the third byte should be fairly easy to understand. This byte represents the velocity at which the note should be played. In our case, we want a velocity of 64, which translates to 40h, and so this is the value which is sent out.

So to recap, we have three bytes for the note on command, "91 3C 40".

```
         Byte 1: 91 == Note on command (on MIDI channel #2)
```

```
        Byte 2: 3C == Note on data (Note to play, C-5)
        Byte 3: 40 == Note on data (Velocity of 64 decimal)
```

Configuring IT's MIDI out data

Keeping the above example in mind, press Shift-F1 to get to IT's MIDI screen and press the 'MIDI Output Configuration' button. This will take you to IT's MIDI out engine. Now, if you examine the 'Note On' field, it reads:

```
      9c n v
```

This can be correlated to our above example of '91 3C 40'. Now, the 'c n v' in the Note On field corresponds to 'channel', 'note' and 'velocity'. Think of them as variables; IT will substitute the appropriate channel, note and velocity values which it encounters in the MIDI instrument information and/or patterndata.

If you now actually defined a MIDI instrument to play on MIDI channel 2, and you played it in a pattern at C-5 with velocity 64, all IT does is read the 'Note On' field from the MIDI configuration screen and substitute '1' for 'c', '3C' (C-5) for 'n' and '64' for 'v'. Therefore, IT will read '9c n v' and replace it with '91 3C 40'.

In any of the MIDI output fields, lowercase letters represent variables (or subsitutions which IT should make) and uppercase letters or numbers are constants which IT writes to the MIDI port directly without any change. Therefore, these fields are case se nsitive--for the note on command, '9c n v' is blatantly different to '9C n v'. 9c represents 'send byte 09 followed by the MIDI channel byte', whereas 9C represents 'send the byte 9C'.

In short, 0-9 and A-F are treated as hexadecimal constants and will be passed through directly. Lowercase letters will be treated as variables and substituted accordingly. Note that variables are regarded as 'full bytes' by themselves and are never part of an actual byte sequence except for the variable 'c', so '9n' is exactly the same as '09 n' or '9 n'; all of them will expand to the sequence '09 <MIDI note byte>'. 'c' is the only value that takes on a nibble (4-byte) value, due to the MIDI protocol definition. This means that 9c will actually become one byte when expanded, with the lower digit representing the channel.

IT MIDI Variables

c: MIDI channel

This is simply the MIDI channel of which the instrument is set to, 0-based. Note that this is the only nibble sized variable.

m: note value (instrument)
A value from 00-7Fh representing the note to be played, where C-5 is 60h. This is the note entered in the pattern, not the translated value.

n: note value (sample)

A value from 00-7Fh representing the note to be played, where C-5 is 60h. This is the note after instrument translations have been applied.

o: Offset value

Extra parameter than can be sent via Oxx commands.

v: velocity

The MIDI velocity of the note.

u: volume

Volume is similar to velocity, except that velocity does not take the volume envelope and fadeout values into account, whereas the 'u' volume variable does.

x: pan set

Sends a MIDI panning value. This does not take into account panning envelopes.

y: calculated pan

Sends a MIDI panning value which does take into account panning envelopes.

a: high byte of bank select
b: low byte of bank select

These commands are only really useful in the bank change field.

z: macro data

(See section on macros for full explanation).

Configuring MIDI Output for Your Keyboard

In the basic IT distribution, the only fields which have any data are 'Note on', 'Note Off' and 'Program Change'. The reason for this is that these are the only commands which are set as standards by MIDI. MIDI commands such as Change Pan, Bank Select , e tc all differ from synth to synth. There's not much which can be done to solve this, you will have to look up your synth's manual to find out the exact MIDI commands it needs to issue a panning change, bank select, etc.

Some values which you may wish to try, however, will be:

```
       Change pan: Bc 0A x
       Bank select: Bc 0 a 20 b
```

These may or may not work. If they do, then great, but if they don't, you will have to actually RTFM *gasp* in order to get these other commands working.

Effect Commands & Macros

Now that IT's MIDI engine is understood and the basis of MIDI communication has been laid down, perhaps the most powerful function of IT's MIDI engine, macros, will be explained.

Standard Effects

Firstly, at the moment there is NO support for standard IT effect commands (such as pitch slide, portamento, vibrato etc) to work via MIDI. This may or may not be implemented in future. Currently, however, if you perform an E01 effect on a note, nothing w ill happen as far as the MIDI aspect of the instrument is concerned.

Macro Effects - SFx

The SFx command, previously used in the .MOD format as "FunkRepeat", has been changed in IT to allow for the functioning of MIDI macros. The unused Zxx command will also now play a part in MIDI functioning.

To understand how this works, it's best to take an example into account. At the beginning of this supplement, the MIDI sequence '91 3C 40' was used which was a Note On, MIDI channel #2 played at C-5 with a velocity of 64. This was represented in IT's MIDI configuration as '9c n v', so it made the appropriate substitutions to '91 3C 40'.

The Macro Setup section of IT's MIDI Output Configuration screen can be used to define your own custom MIDI command/data sequences. These can be absolutely anything you like, from a MIDI SysEx command to a Note On command. In fact, to start off, we'll tak e a Note On sequence as an example and we will attempt to emulate the same '91 3C 40' bytes, except that we'll make this sequence ourselves rather than letting IT do the work for us.

How SFx and Zxx commands relate

Firstly, remember that IT substitutes values when it encounters variables. If you glance at the above section on IT MIDI Variables, you'll notice that the 'z' variable represents macro data. Now that this '91 3C 40' sequence has been driven into our he ads, try setting the SF0 macro field on IT's MIDI Output Configuration screen to '91 3C z'.

Remember that the third byte in the MIDI sequence (40 in our normal example) is the velocity to send with the Note On message. The SF0 macro field you've just defined means that IT will read any Zxx effects and replace the 'z' variable in the SF0 macro with the 'xx' value from the Zxx effect. To enable the macro, simply put in a SF0 along with a Note On in the pattern data. Now, all values from Z00 to Z7F will substitute for 'z' accordingly. So, to show that our SF0 sequence will reproduce the exact same thing as our Note On command:

```
      C-5 01 SF0 (this will play the note on command as usual, and
                  specify that the SF0 macro sequence should be
                  hooked to Zxx effects).
      ... .. ...
      ... .. ...
      ... .. ...
      ... .. Z40 (this will trigger our SF0 sequence with a 'z'
                  value of 40h).
```

The above patterndata should produce a note on event at row 0 in the pattern, and again at row 4. Now, try replacing the Z40 effect with Z7F and IT will substitute 'z' with '7F', or a velocity of 7Fh (127 decimal) in our SF0 sequence. The result should be that you'll hear a Note On with velocity 64 on row 0, and a Note On with velocity 127 on row 4; ie: the second note triggered will be twice as loud. The The sequence that IT will send will be '91 3C 7F'.

SFx commands summary

Our example above of using a 'note on' command sequence for an SFx effect is rather pointless, since IT does this effect itself. However, it has hopefully served its purpose by demonstrating how effects work.

The SFx commands, as you can see, can be redefined to absolutely any MIDI data at all. This can be something simple like a pitch slide, a complex SysEx 'set filter to aftertouch' command, or whatever you like. The possibilities are endless and are only li mited by what your synth can do; IT's SFx/Zxx combination is customisable enough to handle nearly any MIDI data you wish to output.

If you wish to take advantage of these commands, you will have to look up the manual for your synth and get stuck into the MIDI/SysEx section. Please do not come to any IT support people asking for help on this subject because every synth is different.

Z80 -> ZFF commands

The Z80 to ZFF commands are also macro sequences, but they have no 'z' variable to substitute for. They are not 'hooked' to any SFx effects, they are straight, direct macro sequences. For example, if you have a MIDI controllable effects unit (such as an Alesis MidiVerb), you may wish to assign the Z80 command to set a up a certain value for the reverb delay length. Later in the song, you can issue a Z81 command to change the reverb delay or turn it off altogether.

To summarise, the Z80 to ZFF commands are similar to SFx macro sequences, but they do not have any extra parameters (whereas the SFx macro's 'z' variables are controlled by Z00 to Z7F).

Contact Information
===================

If there are any problems with this textfile, email ozone@post1.com or pulse@cyburbia.net.au

```
  __/\___/\_/\____/\____/\ .. . Andre Pang % vault        ...:

 /   /__  /   \_    \_  __) :.. mailto:ozone@post1.com . ....:
(  :  /  (__:  )  ¦  ¦  _)_ : . http://www.mindflux.com.au/ .:
 \___(  ______/¦__;__¦_____¦ :. irc: #trax (irc.neato.org) ..:
      \/ - #ozone
```

Networked Impulse Tracker

What is Networked Impulse Tracker?

Networked Impulse Tracker is simply that - a session of Impulse Tracker where multiple composers can all edit the same song at the same time! It may sound a little bizarre, but networked sessions can be both extremely fun and productive.

Requirements

1. Impulse Tracker, 7 Apr 99 or later

2. Impulse Tracker Network driver file. (*.NET)

3. Some form of network supported by the network driver file.

To Use

To initiate a Network session, Press Shift-ESC. A list of available drivers will be shown. Select one with Enter.

ITIPX.NET

ITIPX.NET is an IPX Network driver for Impulse Tracker. It is recommended that this is used over a LAN (ie. you will need a network card in your computer). Although Kali will work (IPX over the internet), performance will probably be unacceptable for most people through a modem.

The IPX driver *will not be stable* under Win95. Upgrade to Win98.
To install IPX to run under Windows, go to Start Menu->Settings->Control Panel.
Select Network, and under the Configuration Tab, press Add, then select
   Protocol->Microsoft->IPX/SPX Compatible Protocol

The IPX driver will list the available sessions in the left hand box. Select a session to join by pressing Enter.

The Username that the IPX driver transmits is associated with each driver file. The public distribution identifies itself as "Unregistered". Different usernames are availble for US$10 each. Payment can be made via Kagi at http://order.kagi.com/?4ZM

Please specify a username, maximum length 15 characters, or else one will be chosen for you. Updated versions of the driver, if made, will be provided free of charge. However, changing your username will still cost $10.

'Normal' usage of this driver should run quite stably. However, if you try hard to make it crash, I'm sure you will be able to.

Note that connections will be automatically dropped if queued packets fail to be transmitted for more than 10 seconds.

General Notes

Impulse Tracker supports a maximum of 4 users per session. Extra users will be automatically discarded.

Many functions have been disabled under network mode. You will receive warning messages in these cases.

Do *NOT* use hardware mixed drivers for networked sessions. This specifically means the AWE32, GUS and Interwave drivers. Since networked sessions can change samples 'behind your back', these drivers will not update correctly.

Summary Information - Command Line

```
-SFilename.Drv - Set soundcard driver

-Sxx     Quick set sound card
            0 = No Sound
            1 = PC Speaker
            2 = Sound Blaster 1.xx
            3 = Sound Blaster 2.xx
            4 = Sound Blaster Pro
            5 = Sound Blaster 16
            6 = Sound Blaster AWE 32
            7 = Gravis UltraSound
            8 = Interwave IC
            9 = Pro Audio Spectrum
            10 = Pro Audio Spectrum 16
            11 = Windows Sound System
            12 = ESS ES1868 AudioDrive
            13 = EWS64 XL Codec
            19 = Generic MPU401 driver
            20 = Disk writer device

-Axxx    Set sound card's address (hexadecimal)
-D#      Set DMA channel (decimal)
-I##     Set IRQ number (decimal)

-M#####  Set mixing speed (decimal)
-L###    Limit number of channels

-C       Control playback in DOS Shell (with Grey +/-, Right Alt & Right Ctrl)
-F       Disable file-colour distinctions
-K       Exchange F1 and F11 keys
-Px      Pattern memory allocation strategy.
           -P0 = Try to store patterns in conventional memory first, EMS is
                 only used once conventional memory runs out.
                  Not recommended, but those of you who use IT in Windows 3.xx
                  should try this option if you get EMS errors. (I recommend
                  that you don't use IT under Windows 3.xx at all)
           -P1 = Use one block of EMS for all patterndata.
                  This is the most memory efficient of all the pattern
                  storage modes - (this is also the default)
           -P2 = Use EMS blocks for each pattern
                  This is a VERY wasteful but 'safe' memory allocation scheme.
-R       Reverse channels (flip left-right), same as Alt-R on the info page.
-T1      Disable usage time indication
-T2      Enable timeslice release
-V1      Override VGA detection/Matrox detection.
-V2      Force matrox compatibility mode (use with -v1)
-V3      Wait for vertical retraces
-X1      Disable internal MMTSR
-X2      Disable mouse
-X3      Disable drive map detection
-X4      Disable cache file creation
```

Summary Information - Effects, alphabetically

```
Volume Column Effects
 Ax - Volume slide up
 Bx - Volume slide down
 Cx - Fine volume slide up
 Dx - Fine volume slide down
 Ex - Pitch slide down
 Fx - Pitch slide up
 Gx - Portament to
 Hx - Vibrato with speed x

General Effects
Axx - Set speed (set number of frames per row)
Bxx - Jump to order
Cxx - Break to row xx of (next) pattern
Dxy - Volume slide, x=0 down; y=0 up; x=F fine down; y=F fine up
Exx - Pitch slide down by xx
EFx - Fine pitch slide down by x
EEx - Extra fine pitch slide down by x
Fxx - Pitch slide up by xx
FFx - Fine pitch slide down by x
FEx - Extra fine pitch slide down by x
Gxx - Portamento to note with speed xx
Hxy - Vibrato with speed x, depth y
Ixy - Tremor with ontime x, offtime y
Jxy - Arpeggio with halftones x and y
Kxy - Dual command: H00 and Dxy
Lxy - Dual command: G00 and Dxy
Mxx - Set channel volume to xx (0->40h)
Nxy - Channel volume slide, x=0 down; y=0 up; x=F fine down; y=F fine up
Oxx - Set sample offset to xx00h
Pxy - Panning slide, x=0 right; y=0 left; x=F fine right; y=F fine left
Qxy - Retrigger note every y frames with volume modifier x
       Values for x:
         0: (nothing)    4: -8           8: (nothing)    C: +8
         1: -1           5: -16          9: +1           D: +16
         2: -2           6: *2/3         A: +2           E: *3/2
         3: -4           7: *1/2         B: +4           F: *2
Rxy - Tremelo with speed x, depth y
S3x - Set vibrato waveform
S4x - Set tremelo waveform
S5x - Set panbrello waveform
       Waveforms for x in S3x, S4x and S5x:
         0 = Sine                      2 = Square
         1 = Ramp down                 3 = Random
S6x - Pattern delay for x frames
S7x - Instrument functions
       Values for x in S7x:
         0: Past note cut              5: Set NNA to note off
         1: Past note off              6: Set NNA to note fade
         2: Past note fade             7: Turn off volume envelope
         3: Set NNA to note cut        8: Turn on volume envelope
         4: Set NNA to continue
S8x - Set pan position
S91 - Set surround sound
SB0 - Set loopback point
SBx - Loop x times to loopback point
SCx - Note cut after x frames
SDx - Note delay for x frames
SEx - Pattern delay for x rows
SFx - Select parameterised MIDI Macro
T0x - Tempo slide down by x
T1x - Tempo slide up by x
Txx - Set tempo (20h->0FFh)
Uxy - Fine vibrato with speed x, depth y
Vxx - Set global volume to xx (0->80h)
Wxx - Global volume slide, x=0 down; y=0 up; x=F fine down; y=F fine up
Xxx - Set panning position (0->0FFh)
Yxy - Panbrello with speed x, depth y
Zxx - MIDI Macro - check MIDI.TXT
```

Summary Information - Effects, categorically

Note: Not all effects are listed here.

Speed Control

Axx	Set speed
T0x	Tempo slide down by x
T1x	Tempo slide up by x
Txx	Set tempo (20h->0FFh)
S6x	Pattern delay for x frames
SEx	Pattern delay for x rows

Position Control

Bxx	Jump to order
Cxx	Break to row xx of (next) pattern
SB0	Set pattern loopback point
SBx	Loop pattern x times

Volume Control

Ax	Volume slide up
Bx	Volume slide down
Cx	Fine volume slide up
Dx	Fine volume slide down
Dxy	Volume slide, x=0 down; y=0 up; x=F fine down; y=F fine up
Ixy	Tremor with ontime x, offtime y
Mxx	Set channel volume to xx (0->40h)
Nxy	Channel volume slide, x=0 down; y=0 up; x=F fine down; y=F fine up
Vxx	Set global volume to xx (0->80h)
Wxx	Global volume slide, x=0 down; y=0 up; x=F fine down; y=F fine up
Rxy	Tremelo with speed x, depth y
S4x	Set tremelo waveform

Panning Control

Xxx	Set panning position (0->0FFh)
S8x	Set pan position
S91	Set surround sound
Pxy	Panning slide, x=0 right; y=0 left; x=F fine right; y=F fine left
Yxy	Panbrello with speed x, depth y
S5x	Set panbrello waveform

Pitch Control

Exx	Pitch slide down by xx
EFx	Fine pitch slide down by x
EEx	Extra fine pitch slide down by x
Ex	Pitch slide down
Fxx	Pitch slide up by xx
FFx	Fine pitch slide up by x
FEx	Extra fine pitch slide up by x
Fx	Pitch slide up
Gxx	Portamento to note with speed xx
Gx	Portamento to
Hxy	Vibrato with speed x, depth y
Hx	Vibrato with speed x
Uxy	Fine vibrato with speed x, depth y
S3x	Set vibrato waveform
Jxy	Arpeggio with halftones x and y

Summary Information - Pattern Editor 1

```
Data Entry
 Alt-0 -> Alt-9  Set skipvalue to 0-9
 . (period)      Clear field(s)
 1               Note cut (^^^)
 `               Note off (~~~) / panning toggle (in volume column)
 Spacebar        Use last (default) note/instrument/volume/effect/effectvalue

              ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~
              ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~ ~ ~~ ~~ ~ ~ ~ ~~ ~ ~
              ~ ~S~~D~ ~ ~G~~H~~J~ ~ ~2~~3~ ~ ~5~~6~~7~ ~ ~9~~0~ ~
              ~ ~~~~~~ ~ ~~~~~~~~~ ~ ~~~~~~ ~ ~~~~~~~~~ ~ ~~~~~~ ~
              ~ Z~ X~ C~ V~ B~ N~ M~ Q~ W~ E~ R~ T~ Y~ U~ I~ O~ P~
              ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


Pattern selection
 +, -               Next/Previous pattern (*)
 Shift +, -         Next/Previous 4 patterns (*)
 Ctrl +, -          Next/Previous order's pattern (*)

Miscellaneous
 Enter              Get default note/instrument/volume/effect
 '<' or Ctrl-Up     Decrease instrument
 '>' or Ctrl-Down   Increase instrument
 Grey '/'           Decrease octave
 Grey '*'           Increase octave
 ',' (comma)        Toggle edit mask for current field

 Ins/Del            Insert/Delete a row to/from current channel
 Alt-Ins/Del        Insert/Delete an entire row from pattern (*)
 Alt-N              Toggle Multichannel
 2*Alt-N            Multichannel selection menu
 Alt-Enter          Store pattern data
 Alt-Backspace      Revert pattern data (*)
 Ctrl-Backspace     Undo - any function with (*) can be undone.

 Ctrl-F2            Set (multiple) pattern length

Cursor Control
 Up/Down            Move up/down by the skipvalue
 Ctrl-Home/End      Move up/down by 1 row
 Alt-Up/Down        Slide pattern up/down by 1 row
 Alt-Left/Right     Move forwards/backwards one channel
 Ctrl-Left/Right    Move left/right between track columns
 Tab/Shift-Tab      Move forwards/backwards to note column
 PgUp/PgDn          Move up/down by n lines (n=Row hilight major)
 Ctrl-PgUp/PgDn     Move to top/bottom of pattern
 Home               Move to start of column/start of line/start of pattern
 End                Move to end of column/end of line/end of pattern
 Backspace          Move to previous position (accounts for Multichannel)

 Ctrl-C             Toggle centralise cursor option.

Track View Functions
 Alt-T              Cycle current track's view
 Alt-R              Remove all track views
 Alt-H              Toggle track-view divisions
 Ctrl-0             Deselect current track
 Ctrl-1 - Ctrl-5    View current track in scheme 1-5
 Ctrl-Left/Right    Move left/right between track columns

 Left-Ctrl &
  Left-Shift 1-4    Quick setup view scheme (and enable cursor-tracking)

 Ctrl-T             Toggle view-channel cursor tracking
```

Summary Information - Pattern Editor 2

Block functions.

Shift-Movement	Mark block
Alt-B	Mark beginning of block
Alt-E	Mark end of block
Alt-D	Quick mark n/2n/4n/... lines (n=Row Hilight Major)
Alt-L	Mark entire column/pattern
Alt-U	Unmark block/release clipboard
Alt-Q	Raise notes by a semitone (*)
Alt-A	Lower notes by a semitone (*)
Alt-S	Set instrument (*)
Alt-V	Set volume/panning (*)
Alt-W	Wipe volume/panning not associated with a note/instrument (*)
Alt-K	Slide volume/panning column (*)
2*Alt-K	Wipe all volume/panning controls (*)
Alt-J	Volume amplifier (*) / Fast Volume attenuate (*)
Alt-Z	Cut block (*)
Alt-X	Slide effect value (*)
2*Alt-X	Wipe all effect & effect value data (*)
Alt-C	Copy block into clipboard
Alt-P	Paste data from clipboard (*)
Alt-O	Overwrite with data from clipboard (*)
Alt-M	Mix data from clipboard with pattern data (*)
Alt-F	Double block length (*)
Alt-G	Halve block length (*)
Alt-I	Select template mode / Fast volume amplify (*)
Ctrl-J	Toggle fast volume amplification with Alt-J/Alt-I

Playback functions

4	Play note under cursor
8	Play row
Ctrl-F6	Play pattern from current row
Ctrl-F7	Set/Clear playback mark (for use with F7)
Alt-F9	Toggle current channel on/off
Alt-F10	Solo current channel on/off

Impulse Header Layout

```
        0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Ŀ
0000: ~'I'~'M'~'P'~'M'~ Song Name, max 26 characters, includes NULL   ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0010: ~.......................................................~PHiligt~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0020: ~OrdNum ~InsNum ~SmpNum ~PatNum ~ Cwt/v ~ Cmwt  ~ Flags ~Special~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0030: ~GV ~MV ~IS ~IT ~Sep~PWD~MsgLgth~Message Offset ~   Reserved    ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0040: ~ Chnl Pan (64 bytes)...........................................~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ

      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0080: ~ Chnl Vol (64 bytes)...........................................~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ

      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
00C0: ~ Orders, Length = OrdNum                                       ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
xxxx: ~ 'Long' Offset of instruments, Length = InsNum*4 (1)           ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
xxxx: ~ 'Long' Offset of samples headers, Length = SmpNum*4 (2)       ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
xxxx: ~ 'Long' Offset of patterns, Length = PatNum*4 (3)              ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
(1) Offset = 00C0h+OrdNum
(2) Offset = 00C0h+OrdNum+InsNum*4
(3) Offset = 00C0h+OrdNum+InsNum*4+SmpNum*4

  Note that if the (long) offset to a pattern = 0, then the
  pattern is assumed to be a 64 row empty pattern.

PHiliht = Pattern row hilight information. Only relevant for pattern
          editing situations.

Cwt:      Created with tracker.
           Impulse Tracker y.xx = 0yxxh
Cmwt:     Compatible with tracker with version greater than value.
           (ie. format version)
OrdNum:   Number of orders in song.
InsNum:   Number of instruments in song
SmpNum:   Number of samples in song
PatNum:   Number of patterns in song
Flags:    Bit 0: On = Stereo, Off = Mono
          Bit 1: Vol0MixOptimizations - If on, no mixing occurs if
                 the volume at mixing time is 0 (redundant v1.04+)
          Bit 2: On = Use instruments, Off = Use samples.
          Bit 3: On = Linear slides, Off = Amiga slides.
          Bit 4: On = Old Effects, Off = IT Effects
                  Differences:
                 - Vibrato is updated EVERY frame in IT mode, whereas
                    it is updated every non-row frame in other formats.
                    Also, it is two times deeper with Old Effects ON
                 - Command Oxx will set the sample offset to the END
                   of a sample instead of ignoring the command under
                   old effects mode.
                 - (More to come, probably)
          Bit 5: On = Link Effect G's memory with Effect E/F. Also
                      Gxx with an instrument present will cause the
                      envelopes to be retriggered. If you change a
                      sample on a row with Gxx, it'll adjust the
                      frequency of the current note according to:

                        NewFrequency = OldFrequency * NewC5 / OldC5;
          Bit 6: Use MIDI pitch controller, Pitch depth given by PWD
          Bit 7: Request embedded MIDI configuration
                 (Coded this way to permit cross-version saving)

Special:  Bit 0: On = song message attached.
                 Song message:
                  Stored at offset given by "Message Offset" field.
                  Length = MsgLgth.
                  NewLine = 0Dh (13 dec)
                  EndOfMsg = 0

                 Note: v1.04+ of IT may have song messages of up to
                       8000 bytes included.
          Bit 1: Reserved
```

```
          Bit 2: Reserved
          Bit 3: MIDI configuration embedded
          Bit 4-15: Reserved

GV:       Global volume. (0->128) All volumes are adjusted by this
MV:       Mix volume (0->128) During mixing, this value controls
          the magnitude of the wave being mixed.
IS:       Initial Speed of song.
IT:       Initial Tempo of song
Sep:      Panning separation between channels (0->128, 128 is max sep.)
PWD:      Pitch wheel depth for MIDI controllers
Chnl Vol: Volume for each channel. Ranges from 0->64
Chnl Pan: Each byte contains a panning value for a channel. Ranges from
           0 (absolute left) to 64 (absolute right). 32 = central pan,
           100 = Surround sound.
           +128 = disabled channel (notes will not be played, but note
                                    that effects in muted channels are
                                    still processed)
Orders:   This is the order in which the patterns are played.
           Valid values are from 0->199.
           255 = "---", End of song marker
           254 = "+++", Skip to next order
```

Old Impulse Instrument Format (cmwt < 200h)

```
        0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~L·
0000: ~'I'~'M'~'P'~'I'~ DOS FileName (12345678.123)                   ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0010: ~00h~Flg~VLS~VLE~SLS~SLE~ x ~ x ~FadeOut~NNA~DNC~TrkVers~NoS~ x‚~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0020: ~ Instrument Name, max 26 bytes, includes NUL..................‚~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0030: ~.......................................~ x ~ x ~ x ~ x ~ x ~ x‚~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0040: ~ Note-Sample/Keyboard Table, Length = 240 bytes...............‚~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ

      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0130: ~ Volume envelope (200 bytes)..................................‚~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ

                                      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
01F8:                                 ~ Node points (25x2 bytes)......~
                                      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

        Total length of old instrument header is 554 bytes.

      Flg:      Bit 0. On = Use volume envelope
                Bit 1. On = Use volume loop
                Bit 2. On = Use sustain volume loop
      VLS:      Volume loop start (node number)
      VLE:      Volume loop end (node number)
      SLS:      Sustain loop start (node number)
      SLE:      Sustain loop end (node number)
      FadeOut:  Ranges between 0 and 64, but the fadeout "Count" is 512.
                Fade applied when:
                1) Note fade NNA is selected and triggered (by another note)
                2) Note off NNA is selected with no volume envelope
                   or volume envelope loop
                3) Volume envelope end is reached

      DNC:      Duplicate note check (0 = Off, 1 = On)
      NNA:      New note action:
                        0 = Note cut
                        1 = Note continue
                        2 = Note off
                        3 = Note fade

      TrkVers:  Tracker version used to save the instrument. This is only
                used in the instrument files.
      NoS:      Number of samples associated with instrument. This is only
                used in the instrument files.

      Note-Sample/Keyboard Table.
       Each note of the instrument is first converted to a sample number
       and a note (C-0 -> B-9). These are stored as note/sample pairs
       (note first, range 0->119 for C-0 to B-9, sample ranges from
       1-99, 0=no sample)

       Volume envelope: Values from 0->64, 0FFh indicating end of envelope.
       (after which note fade applies)

      Node data: Tick THEN magnitude
```

Impulse Instrument Format

```
        0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Ŀ
0000: ~'I'~'M'~'P'~'I'~ DOS FileName (12345678.123)                   ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0010: ~00h~NNA~DCT~DCA~FadeOut~PPS~PPC~GbV~DfP~RV ~RP ~TrkVers~NoS~ xˆ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0020: ~ Instrument Name, max 26 bytes, includes NUL.................ˎ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0030: ~.......................................~IFC~IFR~MCh~MPr~MIDIBnk~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0040: ~ Note-Sample/Keyboard Table, Length = 240 bytes..............ˎ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ

      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0130: ~ Envelopes...................................................ˎ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
```

```
  NNA = New Note Action
          0 = Cut                 1 = Continue
          2 = Note off            3 = Note fade

  DCT = Duplicate Check Type
          0 = Off                 1 = Note
          2 = Sample              3 = Instrument

  DCA: Duplicate Check Action
          0 = Cut
          1 = Note Off
          2 = Note fade
```

FadeOut: Ranges between 0 and 128, but the fadeout "Count" is 1024
See the Last section on how this works.
Fade applied when:
1) Note fade NNA is selected and triggered (by another note)
2) Note off NNA is selected with no volume envelope
or volume envelope loop
3) Volume envelope end is reached

PPS: Pitch-Pan separation, range -32 -> +32
PPC: Pitch-Pan center: C-0 to B-9 represented as 0->119 inclusive

GbV: Global Volume, 0->128
DfP: Default Pan, 0->64, &128 => Don't use
RV: Random volume variation (percentage)
RP: Random panning variation (panning change - not implemented yet)

MCh = MIDI Channel
MPr = MIDI Program (Instrument)

TrkVers: Tracker version used to save the instrument. This is only used in the instrument files.
NoS: Number of samples associated with instrument. This is only used in the instrument files.

Note-Sample/Keyboard Table.
Each note of the instrument is first converted to a sample number and a note (C-0 -> B-9). These are stored as note/sample byte pairs (note first, range 0->119 for C-0 to B-9, sample ranges from 1-99, 0=no sample)

Envelope layout

```
        Envelopes: 3 structures, first for volume (130h), second for
                   panning (182h), third for pitch (1D4h).

                   Each is structured as such:

        0   1   2   3   4   5   6.......
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Ŀ
xxxx: ~Flg~Num~LpB~LpE~SLB~SLE~ Node points, 25 sets, 75 bytes....~ x ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ

        Flg: Bit 0: Envelope on/off, 1 = on, 0 = off
             Bit 1: Loop on/off, 1 = on, 0 = off
             Bit 2: SusLoop on/off, 1 = on, 0 = off

        Num = Number of node points

        LpB = Loop beginning            SLB = Sustain loop beginning
        LpE = Loop end                  SLE = Sustain loop end

        Node point = 1 byte for y-value
                          (0->64 for vol, -32->+32 for panning or pitch)
                     1 word (2 bytes) for tick number (0->9999)

        Total length of an instrument is 547 bytes, but 554 bytes are
        written, just to simplify the loading of the old format. (Hence
        there are 7 'wasted' bytes per instrument)
```

Impulse Sample Format

```
        0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Ŀ
0000: ~'I'~'M'~'P'~'S'~ DOS Filename (12345678.123)                 ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0010: ~00h~GvL~Flg~Vol~ Sample Name, max 26 bytes, includes NUL.......~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0020: ~.......................................................~Cvt~DfP~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0030: ~ Length          ~ Loop Begin      ~ Loop End        ~ C5Speed       ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
0040: ~ SusLoop Begin ~ SusLoop End   ~ SamplePointer ~ViS~ViD~ViR~ViT~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

The cache file has the following pieces of information added on:

```
        0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Ŀ
0050: ~ File Size     ~ Date  ~ Time  ~Fmt~...........................~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

        Fmt. 0 = unchecked. 1 = directory, 2 = it sample, 3 = st sample
```

```
      GvL:      Global volume for instrument, ranges from 0->64
      Flg:      Bit 0. On = sample associated with header.
                Bit 1. On = 16 bit, Off = 8 bit.
                Bit 2. On = stereo, Off = mono. Stereo samples not supported yet
                Bit 3. On = compressed samples.
                Bit 4. On = Use loop
                Bit 5. On = Use sustain loop
                Bit 6. On = Ping Pong loop, Off = Forwards loop
                Bit 7. On = Ping Pong Sustain loop, Off = Forwards Sustain loop
      Vol:      Default volume for instrument

      Length:   Length of sample in no. of samples NOT no. of bytes
      LoopBeg:  Start of loop (no of samples in, not bytes)
      Loop End: Sample no. AFTER end of loop
      C5Speed:  Number of bytes a second for C-5 (ranges from 0->9999999)
      SusLBeg:  Start of sustain loop
      SusLEnd:  Sample no. AFTER end of sustain loop

      SmpPoint: 'Long' Offset of sample in file.

      ViS:      Vibrato Speed, ranges from 0->64
      ViD:      Vibrato Depth, ranges from 0->64
      ViT:      Vibrato waveform type.
                        0=Sine wave
                        1=Ramp down
                        2=Square wave
                        3=Random (speed is irrelevant)
      ViR:      Vibrato Rate, rate at which vibrato is applied (0->64)

        The depth of the vibrato at any point is worked out in the following
        way:
          Every processing cycle, the following occurs:
                1) Mov AX, [SomeVariableNameRelatingToVibrato]
                2) Add AL, Rate
                3) AdC AH, 0
                4) AH contains the depth of the vibrato as a fine-linear slide.
                5) Mov [SomeVariableNameRelatingToVibrato], AX  ; For the next
                                                                ; cycle.

        For those that don't understand assembly, then the depth is
        basically the running-sum of the rate divided by 256.

        Sample vibrato uses a table 256-bytes long

   Convert - bits other than bit 0 are used internally for the loading
             of alternative formats.
        Bit 0:
         Off: Samples are unsigned   } IT 2.01 and below use unsigned samples
          On: Samples are signed     } IT 2.02 and above use signed samples
        Bit 1:
         Off: Intel lo-hi byte order for 16-bit samples    } Safe to ignore
         On: Motorola hi-lo byte order for 16-bit samples  } these values...
        Bit 2:                                             }
         Off: Samples are stored as PCM values             }
          On: Samples are stored as Delta values           }
        Bit 3:                                             }
```

```
          On: Samples are stored as byte delta values      }
              (for PTM loader)                             }
        Bit 4:                                             }
          On: Samples are stored as TX-Wave 12-bit values  }
        Bit 5:                                             }
          On: Left/Right/All Stereo prompt                 }
        Bit 6: Reserved
        Bit 7: Reserved

   DfP - Default Pan. Bits 0->6 = Pan value, Bit 7 ON to USE (opposite of inst)
```

Impulse Pattern Format

```
        0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~L·
0000: ~Length ~ Rows  ~ x ~ x ~ x ~ x ~ Packed data................  ˆ~
      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ĵ
```

```
Length:   Length of packed pattern, not including the 8 byte header
          Note that the pattern + the 8 byte header will ALWAYS
          be less than 64k
Rows:     Number of rows in this pattern (Ranges from 32->200)
```

Patterns are unpacked by the following pseudocode... (this may look horrible, but in practise, it's just as convenient as the S3M pattern format for playback (but not for display))

```
GetNextChannelMarker:
  Read byte into channelvariable.
  if(channelvariable = 0) then end of row
  Channel = (channelvariable-1) & 63            ; Channel is 0 based.
  if(channelvariable & 128) then read byte into maskvariable
    else maskvariable = previousmaskvariable for current channel

  if(maskvariable & 1), then read note. (byte value)
          // Note ranges from 0->119 (C-0 -> B-9)
          // 255 = note off, 254 = notecut
          // Others = note fade (already programmed into IT's player
          //                     but not available in the editor)

  if(maskvariable & 2), then read instrument (byte value)
          // Instrument ranges from 1->99

  if(maskvariable & 4), then read volume/panning (byte value)
          // Volume ranges from 0->64
          // Panning ranges from 0->64, mapped onto 128->192
          // Prepare for the following also:
          //  65->74 = Fine volume up
          //  75->84 = Fine volume down
          //  85->94 = Volume slide up
          //  95->104 = Volume slide down
          //  105->114 = Pitch Slide down
          //  115->124 = Pitch Slide up
          //  193->202 = Portamento to
          //  203->212 = Vibrato

  Effects 65 is equivalent to D0F, 66 is equivalent to D1F -> 74 = D9F
  Similarly for 75-84 (DFx), 85-94 (Dx0), 95->104 (D0x).

  (Fine) Volume up/down all share the same memory (NOT shared with Dxx
  in the effect column tho).

  Pitch slide up/down affect E/F/(G)'s memory - a Pitch slide
  up/down of x is equivalent to a normal slide by x*4

  Portamento to (Gx) affects the memory for Gxx and has the equivalent
  slide given by this table:

  SlideTable      DB      1, 4, 8, 16, 32, 64, 96, 128, 255

  Vibrato uses the same 'memory' as Hxx/Uxx.

  if(maskvariable & 8), then read command (byte value) and commandvalue
          // Valid ranges from 0->31 (0=no effect, 1=A, 2=B, 3=C, etc.)

  if(maskvariable & 16), then note = lastnote for channel
  if(maskvariable & 32), then instrument = lastinstrument for channel
  if(maskvariable & 64), then volume/pan = lastvolume/pan for channel
  if(maskvariable & 128), then {
          command = lastcommand for channel and
          commandvalue = lastcommandvalue for channel
  }
  Goto GetNextChannelMarker
```

Mathematics

```
Abbreviations:
 FV = Final Volume (Ranges from 0 to 128). In versions 1.04+, mixed output
      devices are reduced further to a range from 0 to 64 due to lack of
      memory.
 Vol = Volume at which note is to be played. (Ranges from 0 to 64)
 SV = Sample Volume (Ranges from 0 to 64)
 IV = Instrument Volume (Ranges from 0 to 128)
 CV = Channel Volume (Ranges from 0 to 64)
 GV = Global Volume (Ranges from 0 to 128)
 VEV = Volume Envelope Value (Ranges from 0 to 64)

In Sample mode, the following calculation is done:
 FV = Vol * SV * CV * GV / 262144               ; Note that 262144 = 2^18
                                                ; So bit shifting can be done.

In Instrument mode the following procedure is used:

 1) Update volume envelope value. Check for loops / end of envelope.
 2) If end of volume envelope (ie. position >= 200 or VEV = 0FFh), then turn
        on note fade.
 3) If notefade is on, then NoteFadeComponent (NFC) = NFC - FadeOut
        ; NFC should be initialised to 1024 when a note is played.
 4) FV = Vol * SV * IV * CV * GV * VEV * NFC / 2^41

Linear slides work like this:
 Final frequency = Original frequency * 2^(SlideValue/768)

(I used a lookup table for the multipliers here)

For command Exx, SlideValue = -4*EffectValue
For command EEx, SlideValue = -EffectValue
For command Fxx, SlideValue = 4*EffectValue
For command FEx, SlideValue = EffectValue

Note that sample vibrato always uses Linear slides.

Notes about effects (as compared to other module formats)

C               This is now in *HEX*. (Used to be in decimal in ST3)
E/F/G/H/U       You need to check whether the song uses Amiga/Linear slides.
H/U             Vibrato in Impulse Tracker is two times finer than in
                any other tracker and is updated EVERY tick.
                If "Old Effects" is *ON*, then the vibrato is played in the
                normal manner (every non-row tick and normal depth)
E/F/G           These commands ALL share the same memory.
Oxx             Offsets to samples are to the 'xx00th' SAMPLE. (ie. for
                16 bit samples, the offset is xx00h*2)
                Oxx past the sample end will be ignored, unless "Old Effects"
                is ON, in which case the Oxx will play from the end of the
                sample.
Yxy             This uses a table 4 times larger (hence 4 times slower) than
                vibrato or tremelo. If the waveform is set to random, then
                the 'speed' part of the command is interpreted as a delay.
```

If you read through this document and there are ANY points which you have troubles with (and have to try out), then let me know - because someone else will have the same questions - and I'd like to make this DOC as easy to understand as possible.

```
For Panning....
 Here's the rough procedure used:

  NotePan = ChannelPan
  if InstrumentPan=On then NotePan = InstrumentPan
  NotePan = NotePan+(InstrumentNote-PPCenter)*PPSeparation/8
```

Pitch Envelopes
 Each value on the envelope equates to half a semitone. This is interpolated 64 times for smooth pitch sliding. Positive values indicate a pitch variation UP of x semitones, negative values indicate a pitch variation down.

General Info

The player in Impulse Tracker 'allocates' channels to notes whenever they are *PLAYED*. In sample mode, the allocation is simple:

```
               Virtual Channel (number) = 'Host' channel (number)
```

In instrument mode, the following procedure is used:

```
 Check if channel is already playing ---Yes--> set 'background' flag on.
             |                                 'Trigger' NNA. If NNA=cut,
             No                                then use this virtual
             |                                 channel.
             |                                          |
             |<------------------ else -----------------/
             |
             v
 Search and find the first non-active virtual channel.
             |
 Non-active channel found? ----Yes----> Use this for playback.
             |
             No
             |
             v
Search through and find the channel of lowest volume that is in the     #
'background' (ie. no longer controlled directly)                         #
             |                                                           #
Background channel found? ----Yes----> Use this for playback.            #
             |                                                           #
             No                                                          #
             |                                                           #
             v                                                           #
Return error - the note is *NOT* allocated a channel, and hence is not   #
played.                                                                  #
```

This is actually quite a simple process... just that it's another of those 'hassles' to have to write...

```
### Note: This is by far the simplest implementation of congestion
          resolution. IT 2.03 and above have a greatly enhanced
          method which more selectively removes the most insignificant
          channel. Obviously, there is no best way to do this - I
          encourage you to experiment and find new algorithms for
          yourself.
```

Internal Tables

```
FineSineData       Label   Byte
        DB       0,  2,  3,  5,  6,  8,  9, 11, 12, 14, 16, 17, 19, 20, 22, 23
        DB      24, 26, 27, 29, 30, 32, 33, 34, 36, 37, 38, 39, 41, 42, 43, 44
        DB      45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 56, 57, 58, 59
        DB      59, 60, 60, 61, 61, 62, 62, 62, 63, 63, 63, 64, 64, 64, 64, 64
        DB      64, 64, 64, 64, 64, 64, 63, 63, 63, 62, 62, 62, 61, 61, 60, 60
        DB      59, 59, 58, 57, 56, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46
        DB      45, 44, 43, 42, 41, 39, 38, 37, 36, 34, 33, 32, 30, 29, 27, 26
        DB      24, 23, 22, 20, 19, 17, 16, 14, 12, 11,  9,  8,  6,  5,  3,  2
        DB       0, -2, -3, -5, -6, -8, -9,-11,-12,-14,-16,-17,-19,-20,-22,-23
        DB     -24,-26,-27,-29,-30,-32,-33,-34,-36,-37,-38,-39,-41,-42,-43,-44
        DB     -45,-46,-47,-48,-49,-50,-51,-52,-53,-54,-55,-56,-56,-57,-58,-59
        DB     -59,-60,-60,-61,-61,-62,-62,-62,-63,-63,-63,-64,-64,-64,-64,-64
        DB     -64,-64,-64,-64,-64,-64,-63,-63,-63,-62,-62,-62,-61,-61,-60,-60
        DB     -59,-59,-58,-57,-56,-56,-55,-54,-53,-52,-51,-50,-49,-48,-47,-46
        DB     -45,-44,-43,-42,-41,-39,-38,-37,-36,-34,-33,-32,-30,-29,-27,-26
        DB     -24,-23,-22,-20,-19,-17,-16,-14,-12,-11, -9, -8, -6, -5, -3, -2

FineRampDownData   Label   Byte
        DB      64, 63, 63, 62, 62, 61, 61, 60, 60, 59, 59, 58, 58, 57, 57, 56
        DB      56, 55, 55, 54, 54, 53, 53, 52, 52, 51, 51, 50, 50, 49, 49, 48
        DB      48, 47, 47, 46, 46, 45, 45, 44, 44, 43, 43, 42, 42, 41, 41, 40
        DB      40, 39, 39, 38, 38, 37, 37, 36, 36, 35, 35, 34, 34, 33, 33, 32
        DB      32, 31, 31, 30, 30, 29, 29, 28, 28, 27, 27, 26, 26, 25, 25, 24
        DB      24, 23, 23, 22, 22, 21, 21, 20, 20, 19, 19, 18, 18, 17, 17, 16
        DB      16, 15, 15, 14, 14, 13, 13, 12, 12, 11, 11, 10, 10,  9,  9,  8
        DB       8,  7,  7,  6,  6,  5,  5,  4,  4,  3,  3,  2,  2,  1,  1,  0
        DB       0, -1, -1, -2, -2, -3, -3, -4, -4, -5, -5, -6, -6, -7, -7, -8
        DB      -8, -9, -9,-10,-10,-11,-11,-12,-12,-13,-13,-14,-14,-15,-15,-16
        DB     -16,-17,-17,-18,-18,-19,-19,-20,-20,-21,-21,-22,-22,-23,-23,-24
        DB     -24,-25,-25,-26,-26,-27,-27,-28,-28,-29,-29,-30,-30,-31,-31,-32
        DB     -32,-33,-33,-34,-34,-35,-35,-36,-36,-37,-37,-38,-38,-39,-39,-40
        DB     -40,-41,-41,-42,-42,-43,-43,-44,-44,-45,-45,-46,-46,-47,-47,-48
        DB     -48,-49,-49,-50,-50,-51,-51,-52,-52,-53,-53,-54,-54,-55,-55,-56
        DB     -56,-57,-57,-58,-58,-59,-59,-60,-60,-61,-61,-62,-62,-63,-63,-64

FineSquareWave     Label   Byte
        DB      128 Dup (64), 128 Dup (0)

EmptyPattern            Label
        DW      64, 64, 0, 0
        DB      64 Dup (0)

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

PitchTable              Label   DWord   ; Values are 16.16 bit
   DW      2048, 0,   2170, 0,   2299, 0,   2435, 0,   2580, 0,   2734, 0 ; C-0
   DW      2896, 0,   3069, 0,   3251, 0,   3444, 0,   3649, 0,   3866, 0 ;>B-0

   DW      4096, 0,   4340, 0,   4598, 0,   4871, 0,   5161, 0,   5468, 0 ; C-1
   DW      5793, 0,   6137, 0,   6502, 0,   6889, 0,   7298, 0,   7732, 0 ;>B-1

   DW      8192, 0,   8679, 0,   9195, 0,   9742, 0,   10321, 0,  10935, 0
   DW      11585, 0,  12274, 0,  13004, 0,  13777, 0,  14596, 0,  15464, 0

   DW      16384, 0,  17358, 0,  18390, 0,  19484, 0,  20643, 0,  21870, 0
   DW      23170, 0,  24548, 0,  26008, 0,  27554, 0,  29193, 0,  30929, 0

   DW      32768, 0,  34716, 0,  36781, 0,  38968, 0,  41285, 0,  43740, 0
   DW      46341, 0,  49097, 0,  52016, 0,  55109, 0,  58386, 0,  61858, 0

   DW      0, 1,      3897, 1,   8026, 1,   12400, 1,  17034, 1,  21944, 1
   DW      27146, 1,  32657, 1,  38496, 1,  44682, 1,  51236, 1,  58179, 1

   DW      0, 2,      7794, 2,   16051, 2,  24800, 2,  34068, 2,  43888, 2
   DW      54292, 2,  65314, 2,  11456, 3,  23828, 3,  36936, 3,  50823, 3

   DW      0, 4,      15588, 4,  32103, 4,  49600, 4,  2601, 5,   22240, 5
   DW      43048, 5,  65092, 5,  22912, 6,  47656, 6,  8336, 7,   36110, 7

   DW      0, 8,      31176, 8,  64205, 8,  33663, 9,  5201, 10,  44481, 10
   DW      20559, 11, 64648, 11, 45823, 12, 29776, 13, 16671, 14, 6684, 15

   DW      0, 16,     62352, 16, 62875, 17, 1790,  19, 10403, 20, 23425, 21
   DW      41118, 22, 63761, 23, 26111, 25, 59552, 26, 33342, 28, 13368, 30

FineLinearSlideUpTable  Label ; Values are 16.16 bit
        DW      0, 1,     59, 1,    118, 1,   178, 1,   237, 1    ; 0->4
        DW      296, 1,   356, 1,   415, 1,   475, 1,   535, 1    ; 5->9
```

```
        DW      594, 1,   654, 1,   714, 1,   773, 1,   833, 1    ; 10->14
        DW      893, 1                                            ; 15

LinearSlideUpTable      Label ; Value = 2^(Val/192), Values are 16.16 bit
        DW      0,      1, 237,   1, 475,   1, 714,   1, 953,   1  ; 0->4
        DW      1194,   1, 1435,  1, 1677,  1, 1920,  1, 2164, 1  ; 5->9
        DW      2409,   1, 2655,  1, 2902,  1, 3149,  1, 3397, 1  ; 10->14
        DW      3647,   1, 3897,  1, 4148,  1, 4400,  1, 4653, 1  ; 15->19
        DW      4907,   1, 5157,  1, 5417,  1, 5674,  1, 5932, 1  ; 20->24
        DW      6190,   1, 6449,  1, 6710,  1, 6971,  1, 7233, 1  ; 25->29
        DW      7496,   1, 7761,  1, 8026,  1, 8292,  1, 8559, 1  ; 30->34
        DW      8027,   1, 9096,  1, 9366,  1, 9636,  1, 9908, 1  ; 35->39
        DW      10181, 1, 10455, 1, 10730, 1, 11006, 1, 11283,1  ; 40->44
        DW      11560, 1, 11839, 1, 12119, 1, 12400, 1, 12682,1  ; 45->49
        DW      12965, 1, 13249, 1, 13533, 1, 13819, 1, 14106,1  ; 50->54
        DW      14394, 1, 14684, 1, 14974, 1, 15265, 1, 15557,1  ; 55->59
        DW      15850, 1, 16145, 1, 16440, 1, 16737, 1, 17034,1  ; 60->64
        DW      17333, 1, 17633, 1, 17933, 1, 18235, 1, 18538,1  ; 65->69
        DW      18842, 1, 19147, 1, 19454, 1, 19761, 1, 20070,1  ; 70->74
        DW      20379, 1, 20690, 1, 21002, 1, 21315, 1, 21629,1  ; 75->79
        DW      21944, 1, 22260, 1, 22578, 1, 22897, 1, 23216,1  ; 80->84
        DW      23537, 1, 23860, 1, 24183, 1, 24507, 1, 24833,1  ; 85->89
        DW      25160, 1, 25488, 1, 25817, 1, 26148, 1, 26479,1  ; 90->94
        DW      26812, 1, 27146, 1, 27481, 1, 27818, 1, 28155,1  ; 95->99
        DW      28494, 1, 28834, 1, 29175, 1, 29518, 1, 29862,1  ; 100->104
        DW      30207, 1, 30553, 1, 30900, 1, 31248, 1, 31599,1  ; 105->109
        DW      31951, 1, 32303, 1, 32657, 1, 33012, 1, 33369,1  ; 110->114
        DW      33726, 1, 34085, 1, 34446, 1, 34807, 1, 35170,1  ; 115->119
        DW      35534, 1, 35900, 1, 36267, 1, 36635, 1, 37004,1  ; 120->124
        DW      37375, 1, 37747, 1, 38121, 1, 38496, 1, 38872,1  ; 125->129
        DW      39250, 1, 39629, 1, 40009, 1, 40391, 1, 40774,1  ; 130->134
        DW      41158, 1, 41544, 1, 41932, 1, 42320, 1, 42710,1  ; 135->139
        DW      43102, 1, 43495, 1, 43889, 1, 44285, 1, 44682,1  ; 140->144
        DW      45081, 1, 45481, 1, 45882, 1, 46285, 1, 46690,1  ; 145->149
        DW      47095, 1, 47503, 1, 47917, 1, 48322, 1, 48734,1  ; 150->154
        DW      49147, 1, 49562, 1, 49978, 1, 50396, 1, 50815,1  ; 155->159
        DW      51236, 1, 51658, 1, 52082, 1, 52507, 1, 52934,1  ; 160->164
        DW      53363, 1, 53793, 1, 54224, 1, 54658, 1, 55092,1  ; 165->169
        DW      55529, 1, 55966, 1, 56406, 1, 56847, 1, 57289,1  ; 170->174
        DW      57734, 1, 58179, 1, 58627, 1, 59076, 1, 59527,1  ; 175->179
        DW      59979, 1, 60433, 1, 60889, 1, 61346, 1, 61805,1  ; 180->184
        DW      62265, 1, 62727, 1, 63191, 1, 63657, 1, 64124,1  ; 185->189
        DW      64593, 1, 65064, 1, 0,      2, 474,   2, 950,   2  ; 190->194
        DW      1427,   2, 1906,  2, 2387,  2, 2870,  2, 3355, 2  ; 195->199
        DW      3841,   2, 4327,  2, 4818,  2, 5310,  2, 5803, 2  ; 200->204
        DW      6298,   2, 6795,  2, 7294,  2, 7794,  2, 8296, 2  ; 205->209
        DW      8800,   2, 9306,  2, 9814,  2, 10323, 2, 10835,2  ; 210->214
        DW      11348, 2, 11863, 2, 12380, 2, 12899, 2, 13419,2  ; 215->219
        DW      13942, 2, 14467, 2, 14993, 2, 15521, 2, 16051,2  ; 220->224
        DW      16583, 2, 17117, 2, 17653, 2, 18191, 2, 18731,2  ; 225->229
        DW      19273, 2, 19817, 2, 20362, 2, 20910, 2, 21460,2  ; 230->234
        DW      22011, 2, 22565, 2, 23121, 2, 23678, 2, 24238,2  ; 235->239
        DW      24800, 2, 25363, 2, 25929, 2, 25497, 2, 27067,2  ; 240->244
        DW      27639, 2, 28213, 2, 28789, 2, 29367, 2, 29947,2  ; 245->249
        DW      30530, 2, 31114, 2, 31701, 2, 32289, 2, 32880, 2 ; 250->254
        DW      33473, 2, 34068, 2                               ; 255->256

FineLinearSlideDownTable Label ; Values are 0.16 bit
        DW      65535, 65477, 65418, 65359, 65300, 65241, 65182, 65359 ; 0->7
        DW      65065, 65006, 64947, 64888, 64830, 64772, 64713, 64645 ; 8->15

LinearSlideDownTable    Label ; Values are 0.16 bit
        DW      65535, 65300, 65065, 64830, 64596, 64364, 64132, 63901 ; 0->7
        DW      63670, 63441, 63212, 62984, 62757, 62531, 62306, 62081 ; 8->15
        DW      61858, 61635, 61413, 61191, 60971, 60751, 60532, 60314 ; 16->23
        DW      60097, 59880, 59664, 59449, 59235, 59022, 58809, 58597 ; 24->31
        DW      58386, 58176, 57966, 57757, 57549, 57341, 57135, 56929 ; 32->39
        DW      56724, 56519, 56316, 56113, 55911, 55709, 55508, 55308 ; 40->47
        DW      55109, 54910, 54713, 54515, 54319, 54123, 53928, 53734 ; 48->55
        DW      53540, 53347, 53155, 52963, 52773, 52582, 52393, 52204 ; 56->63
        DW      52016, 51829, 51642, 51456, 51270, 51085, 50901, 50718 ; 64->71
        DW      50535, 50353, 50172, 49991, 49811, 49631, 49452, 49274 ; 72->79
        DW      49097, 48920, 48743, 48568, 48393, 48128, 48044, 47871 ; 80->87
        DW      47699, 47527, 47356, 47185, 47015, 46846, 46677, 46509 ; 88->95
        DW      46341, 46174, 46008, 45842, 45677, 45512, 45348, 45185 ; 96->103
        DW      45022, 44859, 44698, 44537, 44376, 44216, 44057, 43898 ;104->111
        DW      43740, 43582, 43425, 43269, 43113, 42958, 42803, 42649 ;112->119
        DW      42495, 42342, 42189, 42037, 41886, 41735, 41584, 41434 ;120->127
        DW      41285, 41136, 40988, 40840, 40639, 40566, 40400, 40253 ;128->135
        DW      40110, 39965, 39821, 39678, 39535, 39392, 39250, 39109 ;136->143
        DW      38968, 38828, 38688, 38548, 38409, 38271, 38133, 37996 ;144->151
        DW      37859, 37722, 37586, 37451, 37316, 37181, 37047, 36914 ;152->159
        DW      36781, 36648, 36516, 36385, 36254, 36123, 35993, 35863 ;160->167
        DW      35734, 35605, 35477, 35349, 35221, 35095, 34968, 34842 ;168->175
        DW      34716, 34591, 34467, 34343, 34219, 34095, 33973, 33850 ;176->183
        DW      33728, 33607, 33486, 33365, 33245, 33125, 33005, 32887 ;184->191
        DW      32768, 32650, 32532, 32415, 32298, 32182, 32066, 31950 ;192->199
        DW      31835, 31720, 31606, 31492, 31379, 31266, 31153, 31041 ;200->207
        DW      30929, 30817, 30706, 30596, 30485, 30376, 30226, 30157 ;208->215
        DW      30048, 29940, 29832, 29725, 29618, 29511, 29405, 29299 ;216->223
        DW      29193, 29088, 28983, 28879, 28774, 28671, 28567, 28464 ;224->231
        DW      28362, 28260, 28158, 28056, 27955, 27855, 27754, 27654 ;232->239
        DW      27554, 27455, 27356, 27258, 27159, 27062, 26964, 26867 ;240->247
        DW      26770, 26674, 26577, 26482, 26386, 26291, 26196, 26102 ;248->255
        DW      26008                                                  ; 256

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Effect Info

Here's about all the info I can think of for effects. "Process" variables are variables used internally by effects to control the direction of playback.. This section has not been completed yet.

First, here is the rough flow chart for processing information, it's not fully detailed, but all of the important steps are outlined.

```
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└
       ~ Set note volume to volume set for each channel          ~
       ~ Set note frequency to frequency set for each channel    ~
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    ~
       ~~~~~~~~~~~~~~~~~~~~~~~~~└
       ~ Decrease tick counter  ~        Yes
       ~  Is tick counter 0 ?    ~~~~~~~~~~~~~~~~~~~~~~~~~└
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~                         ~
                    ~                                      ~
                No  ~                ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└
       ~~~~~~~~~~~~~~~~~~~~~~~~~└   ~ Tick counter = Tick counter set        ~
       ~ Update effects for each ~   ~                  (the current 'speed') ~
       ~  channel as required.   ~   ~      Decrease Row counter.             ~
       ~                         ~   ~        Is row counter 0?               ~
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
           ~                                  No  ~          ~
           ~                ~~~~~~~~~~~~~~~~~~~~~~~~         ~ Yes
           ~                ~                                ~
           ~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~└ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└
           ~  ~ Call update-effects for    ~ ~  Row counter = 1               ~
           ~  ~ each channel.              ~ ~                                ~
           ~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~ Increase ProcessRow            ~
           ~                 ~               ~ Is ProcessRow > NumberOfRows?  ~
           ~~~~~~~~~~~~~~~~~~~               ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
           ~                                      Yes ~                  ~ No
           ~         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└   ~
           ~         ~  ProcessRow = BreakRow                        ~   ~
           ~         ~  BreakRow = 0                                 ~   ~
           ~         ~  Increase ProcessOrder                        ~   ~
           ~         ~  while Order[ProcessOrder] = 0xFEh,           ~   ~
           ~         ~                         increase ProcessOrder ~   ~
           ~         ~  if Order[ProcessOrder] = 0xFFh,              ~   ~
           ~         ~                         ProcessOrder = 0      ~   ~
           ~         ~  CurrentPattern = Order[ProcessOrder]         ~   ~
           ~         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~   ~
           ~                               ~                             ~
           ~                               ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
           ~                               ~
           ~         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└
           ~         ~ CurrentRow = ProcessRow                        ~
           ~         ~ Update Pattern Variables (includes jumping to  ~
           ~         ~  the appropriate row if requried and getting   ~
           ~         ~  the NumberOfRows for the pattern)             ~
           ~         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
           ~                               ~
           ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
           ~
       ~~~~~~~~~~~~~~~~~~~└       Yes        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└
       ~ Instrument mode?  ~~~~~~~~~~~~~~~~~~┘ Update Envelopes as required  ~
       ~~~~~~~~~~~~~~~~~~~~~                  ~ Update fadeout as required    ~
           ~                                  ~ Calculate final volume if req ~
           ~ No (Sample mode)                 ~ Calculate final pan if req    ~
           ~                                  ~ Process sample vibrato if req ~
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~└~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       ~ Calculate final volume if required  ~                ~
       ~ Calculate final pan if reqruied     ~                ~
       ~ Process sample vibrato if required  ~                ~
       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~                ~
```

```
          ~                                                     ~
          ~                                                     ~
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                                   ~
                   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~L
                   ~ Output sound!!!                         ~
                   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Axx      Set Tempo

         if (xx != 0) {
                 Maxtick = xx;
                 Currenttick = xx;
         }

Bxx      Jump to Order

         ProcessOrder = xx - 1;
         ProcessRow = 0xFFFE; // indicates new pattern internally for IT...

Cxx      Break to Row

         BreakRow = xx;
         ProcessRow = 0xFFFE;

Dxx      Volume slide down

         if (xx == 0) then xx = last xx for (Dxx/Kxx/Lxx) for this channel.

         Order of testing: Dx0, D0x, DxF, DFx

Dx0      Set effect update for channel enabled if channel is ON.
         If x = F, then slide up volume by 15 straight away also (for S3M compat)
         Every update, add x to the volume, check and clip values > 64 to 64
D0x      Set effect update for channel enabled if channel is ON.
         If x = F, then slide down volume by 15 straight away also (for S3M)
         Every update, subtract x from the volume, check and clip values < 0 to 0
DxF      Add x to volume straight away. Check and clip values > 64 to 64
DFx      Subtract x from volume straight away. Check and clip values < 0 to 0

Hxy      Vibrato

         if (x != 0) {
                 speed = 4*x;
         }
         if (y != 0) {
                 depth = y * 4;
                 if(OldEffects) depth <<= 1;
         }
         Set effect update for channel enabled if channel is ON.
         Goto InitVibrato (explained later)

Ixy      Tremor, ontime x, offtime y

         if (x != 0) {
                 ontime = x;
                 if (oldeffects) ontime++;
         }
         if (y != 0) {
                 offtime = y;
                 if (oldeffects) offtime++;
         }

Nxx      Channel volume slide down

         if (xx == 0) then xx = last Nxx for this channel.

         Order of testing: Nx0, N0x, NxF, NFx

Nx0      Set effect update for channel enabled.
         Every update, add x to the volume, check and clip values > 64 to 64
N0x      Set effect update for channel enabled.
         Every update, subtract x from the volume, check and clip values < 0 to 0
NxF      Add x to volume straight away. Check and clip values > 64 to 64
NFx      Subtract x from volume straight away. Check and clip values < 0 to 0

Uxy      Fine Vibrato

         if (x != 0) {
                 speed = 4*x;
         }
         if (y != 0) {
                 depth = y;
```

```
                if(OldEffects) depth <<= 1;
        }
        Set effect update for channel enabled if channel is ON.
        Goto InitVibrato (explained later)

Wxx     Global volume slide down

        if (xx == 0) then xx = last Wxx for this channel.

        Order of testing: Wx0, W0x, WxF, WFx

Wx0     Set effect update for channel enabled.
        Every update, add x to the volume, check and clip values > 128 to 128
W0x     Set effect update for channel enabled.
        Every update, subtract x from the volume, check and clip values < 0 to 0
WxF     Add x to volume straight away. Check and clip values > 128 to 128
WFx     Subtract x from volume straight away. Check and clip values < 0 to 0

.. sorry this is incomplete..
```

What is VSound?

VSound is a virtual sound driver for Impulse Tracker. It actually uses Microsoft's DirectSound to provide output on ANY soundcard supported by Windows '95 or Windows '98. It does NOT support Windows NT.

It does take a little effort to setup and doesn't perform quite as well as the native drivers, so if you have a SB16, ESS or some other card directly supported by IT, don't bother messing around VSound. If you have a SBPro compatible card, some PCI card or anything else that isn't being used to its maximum capabilities, then give these files a try.

How to use these files

You MUST have DirectX installed. It was written with the DirectX6 SDK, although I'm pretty sure DirectSound 5 is sufficient.

1. Copy ITVSOUND.VXD into your Windows\System directory (normally C:\WINDOWS\SYSTEM)

   Note that the file may NOT show up in Window's Explorer since they have a system extension (.VXD). Use DOS, or enable (unhide) system files in your Explorer configuration.

2. In your Windows directory (normally C:\WINDOWS), edit your SYSTEM.INI file and include the line:
   ```
     device=itvsound.vxd
   ```
   anywhere in the [386Enh] section

3. Reboot your computer

4. Run Server.EXE

5. Run "IT"

Configuring the driver

The driver can be configured by modifying Window's Registry by running "regedit" (which comes with Windows). The variables can be found at:

```
 "KHEY_LOCAL_MACHINE\Software\Jeffrey Lim\Impulse Tracker VSound Server"
```

```
The 4 variables are:
 BufferSize       - The size of the DirectSound buffer in kb.
                    Permitted ranges from 4 to 64, default 24
 BufferThreshold  - The size of the buffer that IT tries to maintain in kb.
                    Permitted ranges from 2 to 32, default 21
 BufferType  - 0 = DualBuffer (BufferSize is logically split into 2 sections)
               1 = QuadBuffer (BufferSize is logically split into 4 sections)
               2 = OctBuffer  (BufferSize is logically split into 8 sections)
                   (recommended, default)
 MixSpeed         - Mixing rate to be used in Hz.
                    Permitted ranges from 11025 to 64000, default 44100
```

You will need to restart the server before these changes take effect.

Known Problems

1. This driver only works with Win95 and Win98, NOT WinNT (I'll try a WinNT driver sometime).

2. There is an obvious latency due to the size of the buffers and the delay that is inherent to DirectSound. Reducing the BufferSize and BufferThreshold values can reduce this latency, however, smaller values can also cause the sound to break up.

    Try using the values:
     1. BufferSize=16, BufferThreshold=14, BufferType=2, MixSpeed=44100

*** PLEASE take the time to check quickly through this document BEFORE ***
*** you write to me. If the answer to your questions lies within here, ***
*** do NOT expect a reply at all. ***

This document is currently incomplete.

1.1 Requirements of IT

NOTE WIN95 USERS -> READ LATER

Impulse Tracker requires a 386+ PC and > 500k of memory. Impulse Tracker uses EMS memory. To setup EMS memory, use the following lines in your CONFIG.SYS file:

```
DEVICE=<path>\HIMEM.SYS
DEVICE=<path>\EMM386.EXE RAM H=255
```

Do *NOT* have "NOEMS" or "FRAME=NONE" on the same line as EMM386.EXE

Also shove these lines in, if they're not already there:

```
DOS=HIGH,UMB <--- just to get yourself a little more memory
STACKS=0,0 <--- Some computers require this to prevent crashing
```

I cannot guarantee that IT will work with QEMM, but Leszek Clapinski wrote to me with this advice (thanks!):
In your config.sys, use:

```
 DEVICE=<path>\QEMM386.SYS DMA=64 HANDLES=255 FORCEEMS
```

Then use "IT -P2"

I recommend that you also devicehigh and Load-high (LH) as many possible drivers, so that you have more conventional memory to play around with.

Win95 Users

If you are a Win95 user, you're probably best off not having HIMEM.SYS *OR* EMM386.EXE in your CONFIG.SYS file. If you *do* have EMM386, make sure you do NOT have "noems" as a parameter.

Win95's internal EMS handling routines *DO* automatically provide the optimum environment for IT, so you should be able to ignore the settings given above.

1.2 Insufficient memory messages

If you get Insufficient Memory messages at the soundcard initialisation, read section 1.3

If you do not have enough conventional memory, the program will exit to DOS almost immediately. If you *JUST* have enough conventional memory, then there may not be enough memory left over to load the sound driver(s) (-> No sound card detected)

1.3 "Mix Data not allocated" messages

All non-wavetable cards require extra *conventional* memory to be allocated in order for them to run appropriately. The amount requires differs between the drivers and depends also on the mixing speed (the higher the mix speed, the more memory required). If you get this message, try to free up some conventional memory.

1.4 Video character's scrambled

Some video cards (esp Matrox cards) did not follow the VGA register standard correctly. Impulse Tracker tries to detecting whether you have such a card, but if this is not successful, run IT /v2 for Matrox compatibility mode.

1.5 Impulse Tracker often hangs

Impulse Tracker may not operate securely in anything other than DOS and Windows 95 (these are the two systems that IT has been extensively tested on) - QEMM/Command shells (eg 4DOS/NDOS) have been known to cause errors in many situations.

Aside from these, if Impulse Tracker hangs on you, please write to me immediately, with a full description of what happens/how you can make it happen. (Including the version of IT that you use!)

If you get a blank screen when you run Impulse Tracker, try using command line parameters to specify your soundcard and port/irq/dma. The autodetect procedures seem pretty reliable, but there's a chance that they may be interfering/interacting with unexpected hardware.

1.6 Files won't load/take a long time to load!

Some music modules are actually compressed with a program called MMCMP. Under normal conditions, these files can be decompressed automatically, under the following situations, they _cannot_:

1) You do NOT have EMM386 loaded - the decompression routines require EMS memory, so if you do not have EMS, you cannot load these files.
2) You are running Impulse Tracker through Windows 3.xx - Windows 3.xx prevents programs from doing certain things... including the setup routines that the decompressor requires to run - so these files cannot be loaded under Windows 3.xx

These files will take longer to load, as they are compressed and are decompressed to disk first.

1.7 Impulse Tracker doesn't support my soundcard!

There may be two reasons for this:
1) Impulse Tracker really doesn't support your soundcard.
2) Impulse Tracker supposedly does support your soundcard but you can't get it to work.

1) Solution: Convince your soundcard manufacturer to send me a sound card to play with AS WELL AS all the programming information. Alternatively, find a soundcard that Impulse Tracker *DOES* support - you can pick up some decent soundcards really cheaply.

2) First of all, check that you have enough memory. If you have a low amount of FreeMem once you load Impulse Tracker, it probably means that there wasn't enough memory to load the sound driver file which will automatically cause a detect failure.

   Unfortunately, not all 100% compatible (esp "100% SBPro compatible") soundcards are REALLY 100% compatible. If the drivers do not detect your soundcard, then try specifying full command line parameters. If it still doesn't work, then I'm sorry - there's nothing I can do about this. Hassle your sound card manufacturer to make decent eqiupment.

   Impulse Tracker uses SB cards in a different *MODE* of playback from most programs so that they are more efficient and also so that they can operate in the background of Windows 95. So just because your soundcard works in another program, it doesn't mean that it's 100% compatible. (this is for all of you who may think "But this card works in other programs, why doesn't it work in IT?")

   For Sound Blaster cards, make sure you have the BLASTER environment variable set in order for IT to detect your card reliably (SB16/AWE32 excluded, as these use hardware routines).

1.8 The sound breaks up/computer slows down when playing songs!

(This section only deals with software mixed cards, ie. almost every soundcard except native GUS, Interwave and EMU8000)

Sound output from the computer requires HEAVY computation - up to 64 thousand calculations per second PER NOTE playing. If your computer is unable to keep up with this, the sound will have very obvious chunks in it and your computer will slow down noticeably.

Solutions:

1) If you're running Impulse Tracker under windows, you may find that running it from a DOS bootup provides a SIGNIFICANT increase the capabilities of your computer in this respect (3x faster in DOS than Win95 for me)
2) Limit the number of notes you can have simultaneously via the command line (/Lxx) - eg. "IT /L32" will limit playback to 32 simultaneous notes maximum.
3) Lower the number of calculations required per second per note. This is done by changing the "mixing speed" via the command line: eg: "IT /m32000" will cause 32000 calculations per second per note to be made. "IT /m22000" will cause 22000 calculations per second per note to be made. Check drivers.txt to find the range of mixing speed values that your soundcard can manage.
4) Get a faster computer :)

1.9 Microsoft Windows and Impulse Tracker

Microsoft Windows 3.xx and Impulse Tracker is a definite no-no. I do NOT guarantee ANYTHING under this configuration. I probably also will not fixup any problems that occur in Impulse Tracker that only occur under Windows 3.xx

Microsft Windows '95 and Impulse Tracker *SHOULD* work fine. If no sound card can be detected under Windows '95, check first that you have no other program using your soundcard (or another DOS window still open that used your soundcard).

Only certain soundcards can play in the background of Windows '95 for technical reasons. Please read the relevant section of DRIVERS.DOC for your soundcard.

Some people have found that Impulse Tracker will hang after a few minutes under Windows '95. Disabling virtual memory may solve this problem. To disable Virtual Memory, right click on My Computer, Properties, Performance, Virtual Memory and check the "disable" box.

2.1 Loading external samples

To load in another sample so that you can use it in your composition, go to the sample-list page (F3), then press "Enter". You will be taken to the "load sample" screen, where you can test out and select samples from a wide variety of formats. This includes:
.IFF, .WAV, .S3I, .ITS, .RAW, and TX Wave .Wxx formats.

You can even load samples DIRECTLY OUT of other modules. In the sample loader, navigate to a drive/directory which contains modules, and you will see that they can opened as 'libraries.' So far, support for external sample loading from modules is available for:
.MOD, .MTM, .S3M, .XM, .669, .PTM, .FAR and of course, .IT

.PAT and .KRZ instruments can also be loaded as sample libraries at the moment.

3.1 What are these CACHE.ITS and CACHE.ITI files?

Whenever you load a sample or instrument, Impulse Tracker has to load all the files to find out their contents, to determine parameters such as sample format, bit fields, etc. CACHE.ITS and CACHE.ITI are files created by Impulse Tracker so that on subsequent usage, this information can be loaded almost instantaneously from a single file, rather than having to reload all the information again.

These may be deleted without disrupting program usage, but they will be recreated when you attempt to load samples/instruments from the directory.

3.2 Distribution Sites

To get the latest versions of IT on the web, check out:

USA Site	- Shawn Mativetsky (Shawn202) http://www.noisemusic.org/it
UK Site	- Andi Simpson (Imminent) http://www.mixbbs.demon.co.uk
Spanish Site	- Javier Gutierrez http://www.musica.org/impulse

```
Music and Tracking Site - Matthias Ziegs (MAZ)
                          http://www.maz-sound.com
IT Resource Central     - Matthew Gardner
                          http://www.unidev.com/~logic/music/it
```

Please don't write to me to become a distribution site - the sites above should be sufficient, and I don't think it is necessary to have BBS distributions since the use of the internet has become so widespread.

To all of the following people listed, I offer my heartfelt thanks.
Impulse Tracker would not be the same without them.

Demosongs for Impulse Tracker

IT1.00 - "Firestorm"	- Chris Jarvis
IT1.01 - "Pale Dreams"	- Chris Jarvis
IT1.03 - "Firepower"	- Chris Jarvis
IT1.05 - "Sidewalk"	- Chris Jarvis
IT1.06 - "Creation of Gaia"	- Sherman Wu (ZaStaR)
IT2.00 - "Fallen World"	- Chris Jarvis
IT2.01 - "A Hidden Fate"	- Chris Jarvis
IT2.04 - "Winter's Dream"	- Andy Sega (Necros)
IT2.08 - "Acid Dreams"	- Liam Widdowson (Legend)
IT2.14 - "Blue Flame"	- Chris Jarvis

Additional Coding for Impulse Tracker

Impulse Tracker Font Customiser	- Sherman Wu (ZaStaR)
Impulse Tracker Text Importer	- Sherman Wu (ZaStaR)
Music Module Compression	- Emmanual Giasson (Zirconia)
Keyboard Configuration files	- Stefan Kucharik (Eliot)

Website Management

USA Site	- Shawn Mativetsky (Shawn202) http://www.noisemusic.org/it
UK Site	- Andi Simpson (Imminent) http://www.mixbbs.demon.co.uk
European Site	- Joost Baaij (CH:ilm) (no longer operational)
Spanish Site	- Javier Gutierrez http://www.musica.org/impulse
Music and Tracking Site	- Matthias Ziegs (MAZ) http://www.maz-sound.com
IT Resource Central	- Matthew Gardner http://www.unidev.com/~logic/music/it

Documentation Help

ASCII Logos	- Ze`ev Nissan (Cruel Creator)
FILE_ID.DIZ	- Ze`ev Nissan (Cruel Creator)
MIDI Output documentation	- Andre Pang (Ozone)

Hardware Thanks

(Sorted alphabetically by company name)

Company: Advanced Micro Devices (AMD)
Received: Interwave Board
Website: http://www.amd.com
Notes: AMD has discontinued the production of the Interwave chips.
          Nonetheless, thank-you to Christopher Cox for his excellent service.

Company: Creative Laboratories
Received: Sound Blaster AWE32
Website: http://www.creaf.com

Company: Hanmesoft / Hoontech, Korea
Received: Sound Track '97
          Sound Track '97 PCI
Website: http://www.hoontech.com
Notes: The Sound Track '97 PCI is the worlds first fully SBPro and WSS

compatible PCI card and has some of the best features ever seen on a soundcard! For bangs per buck, it's hard to beat one of these!

Thanks go to Haejin Park and to Seungho Pak!

Company: Synergy Advanced Technology (Taiwan)
Received: ViperMAX
Website: http://www.synergy.ca/pctoybox
Notes: For a fully compatible Gravis UltraSound card with perfect Sound Blaster compatibility, check out the ViperMAX cards - providing the best of two worlds without the troubles of PnP.

Special thanks go to James Hsu, Synergy Advanced Technology.

Company: TerraTec International
Received: AudioSystem EWS64XL
SoundSystem Maestro 32/96
SoundSystem Maestro 16/96
SoundSystem Gold 16/96
SoundSystem Base1
Website: http://www.terratec.de
Notes: Terratec produces extremely high quality soundcards to cover everyone's needs - check out their high end EWS64XL cards!

Thanks go to the entire Terratec Team, especially Kay Bruns, Wim Roegels and Sascha Kamps - 3 of the team that I've had the pleasure to come in contact with!

Contributions

(sorted in alphabetical order, listing is Full Name / Alias / Country + notes)

Name	Alias	Country	Notes
<Withheld>	Dire	Portugal	
<Withheld>	KjWise	Iceland	
Egor Abramov		Russia	
Habib Al-Assaad			
Tony Allen		UK	MEGA HUGE contribution, thanks!
Simon Altman	P'Wolverine	Australia	
Tal Amir		Israel	
Asbjoern Andersen	Mystical	Denmark	
Steven Anderson		USA	Big contribution, thanks!
Mikael Andersson		Sweden	
Thomas Andersson	Divion	Sweden	
Peter Andries		Belgium	
Jose Angel		Spain	
Tor Erik Arntzen		Norway	
Nicolas Arrouet	Onix4MAN	France	
Patrick Arzul		Sth Africa	Big contribution, thanks!
Tal Asa			
Peter Askel~f		Sweden	
Hans-Joachim Backe		Germany	
Martin Bahner		Norway	
Erick Baker		USA	Big contribution, thanks!
Rodger Ballard	Saqquara	USA	
Brandon Bannerman	Catspaw	USA	
John Barger		USA	MEGA HUGE contribution, thanks!!
Alex Barnell		England	
Matthew Barnes		USA	
Jayson Barrons		USA	
Claus Bartels		Germany	
Jonathan Bartlett			
Michael Baumann		Germany	
David Benjamin		USA	
Brian Bennetts	Daedalus	USA	
Vaughan Bentley		South Africa	
John Bergman		USA	Big contribution, thanks!
Jeff Best		Australia	
Allen Bettilyon	Tremlo	USA	
Bastiaan Bijl	Thanatos	Netherlands	
Mike Blaine			
Christian Bode		Germany	
Lembrecht Bodo		Germany	

Fabian Boes		Germany	
Vicente Beltran Boil	Razz	Spain	
Zozo Bogyo		Hungary	Big contribution, thanks!
Nathan Bonfiglio	Ash		Please contact me!
Eman Borg		Malta	
Gianluca Bove	Yello '73	Italy	
Martin Boverhof		Netherlands	
Alexander Brandon	Siren	USA	
Yannis Brown	Yannis	Australia	
Remko Brugman	Digistorm	Netherlands	
Michael Buchholtz		Germany	Big contribution, thanks!
J~rg Burbach		Germany	
Mike Burrell	MikPos	Canada	
Clay Busker		USA	Big contribution, thanks!
Jeremy Butcher		Netherlands	
Robert Buecker III		USA	
Gunnar Buettner		Germany	
Jim Cairns		Canada	
Rimas Campe		USA	
Mike Cantelon	Foolish Bird	Canada	
Isaac Carrasco		Spain	Big contribution, thanks!
Edward Cashin		USA	
Zach Cappelletti	Shams Kitz	USA	
Michael Carlsson	Silverstance	Sweden	
Christopher Castiglione		USA	
Nilton Catsillo		Chile	
Hector Chang	ZoneSeek	Canada	Big contribution, thanks!
Eric Charlent	Spher-X	France	
Eric Chavanon		France	
Andy Chen		USA	
Charles Cho	Deadsoul	USA	Big contribution, thanks!
Kenny Chou		USA	
Rick Christy	GrymmJack	USA	
Rogier Claessens		Netherlands	Big contribution, thanks!
Jeff Clement		Canada	
David Clipperton		Canada	
Mike Cody		USA	
Shoshi Cohen		Israel	
Caleb Coppock		USA	Big contribution, thanks!
Ben Cormier		Canada	
David Cornish		Australia	Big contribution, thanks!
Martin Cosgrave		UK	
Antony Cowderoy		UK	
David Cox-Espenlaub		USA	
Matt Cramer		USA	
Jim Crawford	Pfister	USA	
Dan Cunningham	Pentatonic	USA	Big contribution, thanks!
Daniel Cunningham		Ireland	
David Cuny		USA	
Nicholas Dahlin		Denmark	
Chad Dahlquist			
Christopher Daniel		USA	
Tristan Daniel	Jesus2099	France	
Thomas Daniels III		USA	MEGA HUGE contribution, thanks!
Ben Dany			
Tulio Guimaraes da Silva		Brazil	HUGE contribution, thanks!
Dave Davis		USA	
Tomer Dayan		Israel	
Fabio De Araujo Neves		Brazil	
Benoit De Greift	Eagle	Belgium	
David Dean	Sector4	Australia	
Pierre Decourcelles		France	Big contribution, thanks!
Barthelemy Defossez		France	
Guy Detienne		Belgium	
John Dietzel		USA	
John Di Giacomo		USA	
Luciano Di Lucrezia	Spectrum		Big contribution, thanks!
Nick Dinges		Germany	
Robin Dittwald	Satixu	Germany	
Siem Doodeman		Netherlands	
Asaf Dor			
Stephen Dredge		Australia	
Andrew Dun		Australia	
Dave Dunger		Australia	
Andrew Durk		USA	
Florian Dvorski	Thunk	Germany	
Brendan Ebel		USA	
Adam Ebringer		Australia	
Michael Edwards		UK	
John Ehmann		Canada	
Richard Eijkenbroek		Netherlands	
Michael Elis		Isreal	
Garrett Ellis		USA	

Irad Eshel		Israel	
Scott Esposto		USA	
Alexander Ewering	Internal	Germany	
Gary Feinmesser		England	
Nick Feldman		UK	
Jim Fergusson		USA	
Robin Fernandes		France	
Dan Fetherstonhaugh		England	
Reuben Firmin	Rubz	Scotland	Big contribution, thanks!
Kyle Fischer		USA	
Edward Flick		USA	
Brad Folkens		USA	
Adam Frank		USA	
Andrew Franks	Derelict	USA	
Graham Freeman		Australia	
Calvin French		Canada	
Joseph Freund		Australia	
David Friberg		USA	
Matt Friedly	Subliminal	USA	
Richard Funke		Norway	
James Furness		UK	
Robert Gage		Australia	
Miles Gannett		USA	
Peter Gaywood		UK	
Christopher General		Canada	
Giovanni Giampieri		Italy	Big contribution, thanks!
Slavko Glamocanin		Slovenia	
Owen Godwin		USA	
Markus Goetz	Cookie Jar	Germany	Big contribution, thanks!
John Goforth		USA	
David Goodale		Canada	
Ferdinand Gozum	DJ	USA	
Jeff Graham		USA	HUGE contribution, thanks!
Joseph Graham	BrianXavier	USA	
Micah Greenlay		USA	
Lorenzo Grifi		Italy	
Sebastian Grillmaier	Wayfinder	Germany	
Moritz Grimm	Maxx	Germany	
Karsten Grombach		Germany	Big contribution, thanks!
Ariel Gross	Stalker	USA	HUGE contribution, thanks!
Sylvian Guiraud		France	Big contribution, thanks!
Joe Hahn		USA	
Peter Hajba	Skaven	Finland	
Doris Hamburger		USA	
Tim Hamers		Netherlands	
David Hamilton		England	
Eric Hamilton	Dilvish	USA	Big contribution, thanks!
John Harris		USA	
Todd Hartley	Tronster	USA	
Ian Haskin	SiN	Canada	
John Hastie		USA	
Joshua Hastey	Ith	USA	
Thomas Havelka		USA	
David Hays	D-Range	USA	
Axel Hedfors		Sweden	
Brett Helgeson		Australia	Big contribution, thanks!
Greg Heo		Canada	Big contribution, thanks!
Liam Hesse	Lemm	England	
Anthony Hicks		USA	
Robert Hilpert		Germany	MEGA HUGE contribution, thanks!
Erik Hjelmvik	Tweety	Sweden	
Jason Hlisic	isotone	Australia	Big contribution, thanks!
John Hobson		USA	
Rune Holm	TitanStar	Norway	
Krystal Holstein		USA	Big contribution, thanks!
Alexandre Holzhey		Brazil	
Aake Honkaniemi	PAH	Finland	Big contribution, thanks!
Raymon Hoving		Finland	
Troy Howard		USA	Big contribution, thanks!
Ben Howell	Rhythm	USA	
Shu Hung		Canada	Big contribution, thanks!
Ryan Hunt	Pinion	USA	
David Hunter		USA	
Steve Hunter		UK	Big contribution, thanks!
Samuel Hurst		Australia	
Kohan Ikin		Australia	
Sam Izzo	Jestyr	Australia	
Shane Jackson	MetaMan	Canada	
Keith Jagielski		USA	
Daniel Janiak	Redrick	Slovakia	Big contribution, thanks!
Marko Janssen		Netherlands	
Simon Jarosch	Vivid	Germany	
Chris Jarvis		Australia	

Dave Jeavons	Realclean	Australia	HUGE contribution, thanks!
James Jeen		Australia	
Joshua Jersild		USA	
Jan Jirak		Czechoslovakia	
Robert Johnson	Xenopraxis		
Mischa Jonker		Netherlands	
Ben Just		Australia	Big contribution, thanks!
Jaakko Kaivosoja		Finland	
Steffen Kamprath		Germany	
Frank Kane		USA	
Ben Kapper		Netherlands	
Ilpo Karkkainen	Griffin	Finland	
Tero Karkinen		Finland	
Thomas Karolczak	Mindflyer XS		
Cole Kelley		USA	
Keith Kelly	Syrinx	USA	
Mehran Khalili	Screamager	Luxembourg	
Leonard Khiroug		USA	
Richard Kidwell		USA	
Thomas Kim		USA	
David Klande		Germany	
Martin Kleinman	Brain	Netherlands	
Martin Klossek		Germany	
Bart Knol	Armadon	Netherlands	
Stephen Knowles		USA	
Odd Henry Knutsen		Netherlands	
Henning Koch		Germany	
Alexander Koenig		Germany	
David Kondrad		USA	
Zaf Korbetis		Australia	Big contribution, thanks!
Evan Korzon		USA	
Christopher Kowalski		USA	
Kristof Kowalski	Cyntax	Australia	
Kevin Krebs		Canada	
Bert Kroes		Netherlands	
Peter Kunath		Germany	MEGA HUGE contribution, thanks!
Rich La Bonte	Flatrich	USA	
Michael Ladanyi		Canada	
Brian J LaMattina		USA	
Perttu Lamminm~ki		Finland	
William Lamy	willbe	France	
Adam Langdon-Thomas		UK	
Janet Lankester		USA	
Patrick Lea		Australia	
Jon Leahy		UK	HUGE contribution, thanks!
Steven LeBeau		USA	
Eliot Lee	PrOtoCoL	USA	Big contribution, thanks!
Nicolas Leveille	Knos	France	
Buzz Libre		USA	
Rik Ling		USA	
Randy Locklair		USA	MEGA HUGE contribution, thanks!
Michael Loftus		USA	MEGA HUGE contribution, thanks!
Kris Long		USA	
Jerome Majewski		USA	HUGE contribution, thanks!
Sami M~kinen		Finland	
Don Manton		Canada	
Peter Mares		South Africa	
Anthony Marin	Verminator	USA	
Norman Mark	Nothing More	Canada	
Alexander Martin		Germany	
Mike Ma'rton		Hungary	
Akos Matte	Bat & Cyborg	Hungary	
Lucas Higa Mattsson		Sweden	
Nicola McAleer		Ireland	
Brad McKinnon	EtherWizard	USA	Big contribution, thanks!
Ray McManus		USA	
Keith McNally		Canada	
Glen McNiece		Australia	
Vincent Meijer		Netherlands	
Mitchell Menghi		Australia	
David Menkes	Behemoth	USA	
Mike Merker		Canada	
Jonathan Mesiano-Crookston		Canada	
Christopher Micali	Zephyr	USA	
John Michael-Lloyd		USA	Big contribution, thanks!
Chris Michetti		Canada	
Pekka Mikkola		Finland	
Alex Milla		Spain	
Matthew Miller		Australia	HUGE contribution, thanks!
Tracey Miller		Australia	Big contribution, thanks!
Matthew Mitchell		USA	Big contribution, thanks!
Chun-Yan Miu		Hong Kong	
Mauro Molinari	DjM	Italy	

Lutz M~ller		Germany	
Jo Moore		USA	
Freddy Mousseau		France	
Marten Mons		Netherlands	
F. Edmund Mueller		USA	
Markus Mueller	G42	Germany	Big contribution, thanks!
Paul Munson		Canada	
Aaron Murray		USA	
Eric Nadeau		Canada	
Riley Nagler		USA	
Tirone Nel		South Africa	
Frank Nentwich		Germany	
Mikkel Nordberg		Denmark	HUGE contribution, thanks!
Gary Norris		USA	
Fumihiro Odaki		Japan	
Michiyasu Odaki		Japan	
John O'Laughlin		USA	
Anthony Oetzmann	AiRON	Germany	
Xavier Orengo		USA	
Frido Otten	Cybertron	Netherlands	Big contribution, thanks!
Olli Paasovaara		Finland	
Robert Pain		UK	
Jonathan Pak	Maelstrom	Australia	
Andre Pang	Ozone	Australia	
Adam Parker		USA	
Jourden Parks	AllenKray	USA	
Kiran Patil		USA	Big contribution, thanks!
Stefan Pavlov			
Christian Pfaff		Germany	Big contribution, thanks!
Jean-Luc Pedneault		Canada	
Nick Pelling		UK	
P-J Perrussel-Morin	CyberNitrox	France	Big contribution, thanks!
Chaim Peter-Chester		USA	
Claus Peterson		Denmark	Big contribution, thanks!
Felix Petrescu	Waka X	Romania	
Henri Pihkala		Estonia	
Jason Phelps		USA	Big contribution, thanks!
Miklavz Pirnat			
Eddie Plaskur	Mexican Beans	Australia	
Harout Pogossian	HP		
Eric Pomerleau	O'Realitea	Canada	
Pascal Q. Porcupine	Spikey	USA	Big contribution, thanks!
William Price		USA	
Alexander Rahm		Germany	HUGE contribution, thanks!
Hazhir Ranjram	Balk	USA	
Alessandro Rascazzo	Antitesi	Italy	
Pasi Rastas		Finland	
Justin Ray	Zinc	Canada	
Tobias Reckhard	Jester	Germany	
Jimmy Redfern	Jimmy	Ireland	
Justin Reid	Stein	USA	
Viktor Rez		Hungary	
Owyn Richen			
Dan Richters		USA	Big contribution, thanks!
Jay Ridley		Australia	
Michael Riegel		USA	
Todd Rieger		USA	Big contribution, thanks!
James Rimmer		USA	
Brendan Robert		USA	
Jeff Robinson		USA	
Josh Rodman	K8to	USA	MEGA HUGE contribution, thanks!
Jens Roeben		Germany	
Nick Rose	Phantasm	Canada	
Ryan Ross		Canada	
Benjamin Rumbaugh		USA	
Keijo Ruonamaa	Gze	Finland	
Geert Rutten	Loonatic	Netherlands	
Hannu Salonen	Salomon	Finland	
Kevin Salt		Netherlands	
Mark Sanders		USA	
Denis dos Santos		Brazil	
Adi Sapir	Doc	Israel	
Janne Savolainen		Finland	
Ben Saylor		USA	Big contribution, thanks!
Konrad Schandera	iCEWiND	GERMANY	
H.R. Scheper-Keuter	Josuf	Netherlands	
Gerben Schmidt		Netherlands	
Martin Schmidt		Germany	
Daniel Schwab		Germany	
Wolfgang Schwarz		Germany	Big contribution, thanks!
Matthew Scott		USA	
Kristian Sergiejew		Poland	
Saurin Shah	Nebula	USA	

Jarrod Sharp		Australia	
Dan Shaw		USA	
Ryan Shaw		USA	
Shaul Shentai		Israel	HUGE contribution, thanks!
Philip Shipley		USA	
Oren Shomron		Isreal	
Josh Silvey		USA	
Morten Skarstad		Norway	
Peter Skeide		Norway	MEGA HUGE contribution, thanks!
Steven Slater		Canada	
Martijn Sneijder		Netherlands	
Trond Smevik	Poke	Norway	
Greg Smith		Canada	
Zachary Smith	Dr. Zachary	USA	
Christoph S~llner	AGENT S	Germany	
Scott Sorenson		USA	
Nicolas Soudee	Zoner	USA	Big contribution, thanks!
Michael Soutar	MSoutar	Australia	
Kasper Souren		Netherlands	
Patrick Stacey		Australia	
Nick Stanfield		UK	
Andrew Ryan Stinnett		USA	
Ian Stocker	RabiteMan	USA	Big contribution, thanks!
Mark Straver	Moonchild	Netherlands	
Jer Sypult		USA	
Charles Tabourot		France	
Ravon Tamar	RavEon	Israel	
Patrice Tarabbia	Mercure	France	
Jason Le Sueur Tatum		USA	Big contribution, thanks!
Thor Teague		USA	
Michael Teehan		USA	Big contribution, thanks!
Michel ten Voorde		Netherlands	
Franck Theuex		France	Big contribution, thanks!
Michael Thomas		USA	
Sebastian Thomas		Scotland	
Jaymz Thompson			
Philip Thompson		UK	
Franz Thues		Germany	
Gabriele Tittarelli	T.S.P.	Italy	
Jason Tracer	Electric Keet	USA	
Nash Trajkowski		Australia	
Eric Tremblay	DeltaX	Canada	
Brad Turcotte		Canada	
Michael Twarkowsky		Germany	
Asbjorn Ulsberg		Norway	
Martin Underwood		UK	
Jani V~is~nen		Finland	
Mathew Valente	TSSF	Canada	
David Van Dromme	Stormlord	Belgium	
Ryan Van Eerdewijk		Canada	
Marti Van Lin			
Erik Van Hengstum		Netherlands	
Ernst Van Rossum		Netherlands	
Filip Van Schoor	Cantaloup	Belgium	
Jan Van Stiphout			
Maarten Van Strien	Crystal Score	Netherlands	
Jos vd Geest		Netherlands	
Oscar Vela		Spain	
Markus Visti		Finland	
Ganesh Viswanathan	Genosha	India	
Benjamin Vogt		Australia	
Edwin Volkmer		Netherlands	
Vincent Voois	Vv	Netherlands	MEGA HUGE contribution, thanks!
Fredrik Von Braun		Sweden	
Arno Vryman		Netherlands	
Petri Vuorio		Finland	
Ben Waddington		Australia	Big contribution, thanks!
Gerd Wagner		Germany	
Chris Wallace	CTS	USA	
Thomas Walter	Subsonic	Australia	
Adrian Ward		England	HUGE contribution, thanks!
Timothy Weller	Flukey	USA	
Bill Wells		USA	
Mike Wells		USA	
Inyoung Whang		USA	
Jemi White		Australia	
Brian Wickman	CD	USA	
Liam Widdowson	Legend	Australia	
Gene Wie	Psibelius	USA	Big contribution, thanks!
David Wiernicki	Perisoft	USA	Big contribution, thanks!
John Williams	Jackal	USA	
John Wilson		USA	
Tobias Wilton		Sweden	Big contribution, thanks!

Mathias Wintzer			
Marco Wotschadlo		Germany	
Doug Wright	Netriangle	USA	
Fuming Wu	Adi	Taiwan	
Takeshi Yamamoto		Japan	Huge contribution, thanks!
Shane Yates	AlphaRISC	Australia	
David Zearing		USA	
Daniel Zegiel		USA	
Matthias Ziegs	MAZ	Germany	

... where's your name? :)