Standard commands (.MOD &.XM)

- `0xy` Arpeggio
- `1xx` Portamento up
- `2xx` Portamento down
- `3xx` Portamento to note
- `4xy` Vibrato
- `5xy` Portamento to note with volume slide
- `6xy` Vibrato with volume slide
- `7xy` Tremolo
- `8xx` Set note panning position
- `9xx` Sample offset
- `Axy` Volume slide
- `Bxx` Jump to order
- `Cxx` Set note volume
- `Dxx` Pattern break
- `Exy` Subcommands:
  - `E0x` Amiga LED Filter toggle *
  - `E1x` Fine portamento up
  - `E2x` Fine portamento down
  - `E3x` Glissando control **
  - `E4x` Vibrato control **
  - `E5x` Set note fine-tune
  - `E6x` Pattern loop
  - `E7x` Tremolo control **
  - `E8x` Set note panning position ***
  - `E9x` Re-trigger note
  - `EAx` Fine volume slide up
  - `EBx` Fine volume slide down
  - `ECx` Note cut
  - `EDx` Note delay
  - `EEx` Pattern delay
  - `EFx` Funk it! *
- `Fxx` Set song speed/BPM

Extended commands (.XM only)

- `Gxx` Set global volume
- `Hxy` Global volume slide
- `Kxx` Key-off
- `Lxx` Set volume envelope position
- `Pxy` Panning slide
- `Rxy` Re-trigger note with volume slide
- `Txy` Tremor
- `Xxy` Extra fine portamentos:
  - `X1x` Extra fine portamento up
  - `X2x` Extra fine portamento down

Volume column commands (.XM only)

- *xx* Set note volume
- *+x* Volume slide up
- *-x* Volume slide down
- *Dx* Fine volume slide down (displayed as ▼x)
- *Lx* Panning slide left (displayed as ◂x)
- *Mx* Portamento to note
- *Px* Set note panning position
- *Rx* Panning slide right (displayed as ▸x)
- *Sx* Set vibrato speed
- *Ux* Fine volume slide up (displayed as ▲x)
- *Vx* Vibrato

*) Not implemented, no plans to support
**) Not implemented yet, will be required for feature completeness
***) Not supported on Amiga nor in FT2, effect relocation (8xx, Px) advised

0xy Arpeggio

Syntax:

0

x = semitone offset

y = semitone offset

Example:

```
C-4 ·1 ·· 037
··· ·· ·· 037
··· ·· ·· 037
··· ·· ·· 037
```

Explanation:

Arpeggio quickly alters the note pitch between the base note (C-4) and the semitone offsets *x* (3 = D#4) and *y* (7 = G-4). Each pitch is played for the duration of 1 tick. If speed is higher than 3 (meaning there are more than 3 ticks per row), the sequence is looped.

ProTracker 2/3

Base note is played for tick 0, then the semitone offset *x* for tick 1, then semitone offset *y* for tick 2.

Fasttracker II

Base note is played for tick 0, then the semitone offset *y* for tick 1, then semitone offset *x* for tick 2.

In MilkyTracker you don't have to and indeed you CAN'T enter the effect digit 0. Just start with the parameter digits and the effect digit will be filled in.

Notes:

Doesn't have effect memory and cannot be used without parameters.

In Fasttracker II, arpeggio logic fails when song speed is 16 (0x10) or higher. Using arpeggio at such speeds may cause unpredictable results across different players.

Tips:

When both effect parameters are used, it is wise to use a song speed value divisible by 3 in order that the arpeggio sequence can loop smoothly.

1xx Portamento up

Syntax:

1

xx = portamento speed

Example:

```
C-4 ·1 ·· 103
··· ·· ·· 103
··· ·· ·· 103
··· ·· ·· 103
```

Explanation:

Portamento is used to slide the note pitch up or down. The higher the *xx*, the faster it goes. Effect is applied on every tick.

Notes:

Amiga frequencies

The slide speed also depends on the sample frequency.

ProTracker 2/3

Doesn't have effect memory and cannot be used without parameters.

2xx Portamento down

Syntax:

2

xx = portamento speed

Example:

```
C-4 ·1 ·· 203
··· ·· ·· 203
··· ·· ·· 203
··· ·· ·· 203
```

Explanation:

Works similarly to *1xx* portamento up, only bending note pitch down instead of up.

Notes:

ProTracker 2/3

Doesn't have effect memory and cannot be used without parameters.

3xx Portamento to note

Syntax:

3

xx = portamento speed

Example:

```
C-4 ·1 ·· ···
E-4 ·1 ·· 304
··· ·· ·· 300
··· ·· ·· 310
```

Explanation:

This portamento command bends the already playing note pitch towards another one, entered with the *3xx* command. In the example, C-4 is bent towards E-4 at portamento speed *04* which isn't fast enough to reach the E-4 pitch during the two rows at the default song speed (6/125). However, *310* on the following row continues the portamento and being much faster, achieves the target E-4 pitch.

4xy Vibrato

Syntax:

`4`

x = speed

y = depth

Example:

```
C-4 ·1 ·· 481
··· ·· ·· 402
··· ·· ·· 400
··· ·· ·· 460
```

Explanation: Vibrato alters note pitch up and down in the maximum range of a full tone. After the initial *xy* pair, parameters can be set individually. The pitch is reset when the command is discontinued.

5xy Portamento to note with volume slide

Syntax:

`5`

x = volume slide up speed

y = volume slide down speed

Example:

```
C-4 ·1 ·· ···
E-4 ·1 ·· 304
··· ·· ·· 504
··· ·· ·· 504
```

Explanation: Performs portamento to note with parameters initialized with *3xx* or *Mx* while sliding volume similarly to *Axy volume slide*.

ProTracker 2/3

Notes:

Doesn't have effect memory for volume slide speeds, *500* works identically to *300*.

6xy Vibrato with volume slide

Syntax:

`6`

x = volume slide up speed

y = volume slide down speed

Example:

```
C-4 ·1 ·· 481
··· ·· ·· 601
··· ·· ·· 600
··· ·· ·· 6C0
```

Explanation: Performs vibrato with parameters initialized with *4xy* or *Sx*+*Vx* while sliding volume similarly to *Axy volume slide*.

ProTracker 2/3

Notes:

Doesn't have effect memory for volume slide speeds, *600* works identically to *400*.

7xy Tremolo

Syntax:

7

x = speed

y = depth

Example:

```
C-4 ·1 ·· 787
··· ·· ·· 700
··· ·· ·· 7C0
··· ·· ·· 700
```

Explanation: Tremolo alters note volume up and down. After the initial *xy* pair, parameters can be set individually. The volume is not reset when the command is discontinued.

8xx Set note panning position

Syntax:

8

xx = panning position

Example:

```
C-4 ·1 ·· 880
··· ·· ·· 8A0
··· ·· ·· 8C0
··· ·· ·· 8F0
```

Explanation: Sets the note stereo panning from far left *00* to far right *FF* overriding sample panning setting.

Notes:

ProTracker 2/3

- On Amiga, the 4 MOD channels are hard panned left, right, right and left by hardware, no use panning manually there.

Fasttracker II

Panning envelopes operate relative to the set position.

9xx Sample offset

Syntax:

9

xx = sample offset

Example:

```
C-4 ·1 ·· ···
··· ·· ·· ···
C-4 ·1 ·· 908
··· ·· ·· ···
```

Explanation: The sample that the note triggers is played from offset *xx*. The offsets are spread 256 samples apart so *908* skips the first (0x8*256=) 2048 bytes of the sample and plays it on from there. This means that the furthest point *9xx* can reach is (0xFF*256 =) 65280 bytes into the sample.

Tips: Resampling a loop to exactly (0x10000=) 65536 bytes gives you the highest possible level of control over the sample.

Axy Volume slide

Syntax:

`A`

`x` = volume slide up speed

`y` = volume slide down speed

Example:

```
C-4 ·1 ·· A04
··· ·· ·· A04
C-4 ·1 ·· A0F
··· ·· ·· A80
```

Explanation: Slides note volume up/down at speed `x`/`y` depending on which parameter is specified. Effect is applied per tick so song speed value acts as a multiplier.

- Parameters `x` and `y` should NOT be used at the same time, doing so almost guarantees unpredictable results across different players.

Notes:

ProTracker 2/3

Doesn't have effect memory and cannot be used without parameters.

Bxx Jump to order

Syntax:

`B`

`xx` = song position

Example:

```
C-4 ·1 ·· ···
··· ·· ·· ···
··· ·· ·· ···
··· ·· ·· B04
```

Explanation: Immediately breaks the current pattern and jumps to order `xx` in the pattern order table (POT).

Tips: Can be used to divide a song into separate looping sections effectively creating multiple songs using the same set of instruments. Such modules can be used in games and such where the sections can be triggered dynamically by program events.

Cxx Set note volume

Syntax:

`C`

`xx` = volume

Example:

```
C-4 ·1 ·· ···
··· ·· ·· C10
··· ·· ·· C40
··· ·· ·· C00
```

Explanation: Sets the note volume `00 – 40` overriding sample volume setting.

Notes:

Fasttracker II

Volume envelopes operate relative to the set volume.

Dxx Pattern break

Syntax:
`D`
`xx` = row number on next pattern

Example:
```
C-4 ·1 ·· ···
··· ·· ·· ···
··· ·· ·· ···
··· ·· ·· D04
```

Explanation: Breaks the current pattern and jumps to row `xx` on the next pattern.

Notes: Unlike with the majority of effect parameters, here `xx` is a decimal value rather than hexadecimal. Hexadecimal values are accepted but the first digit is still interpreted as decimal so it's best to avoid hex this time.

The highest row number you can jump to is 63.

E1x Fine portamento up

Syntax:
`E1`
`x` = portamento speed

Example:
```
C-4 ·1 ·· E11
··· ·· ·· E12
··· ·· ·· E13
··· ·· ·· E14
```

Explanation: Works similarly to *1xx portamento up*, only the slide is a lot finer because the effect is applied only once per row.

E2x Fine portamento down

Syntax:
`E2`
`x` = portamento speed

Example:
```
C-4 ·1 ·· E11
··· ·· ·· E12
··· ·· ·· E13
··· ·· ·· E14
```

Explanation: Works similarly to *2xx portamento down* bending note pitch down, only the slide is a lot finer like with `E1x`.

E3x Glissando control

Syntax:
`E3`
`x` = glissando control toggle on/off

Example:
```
C-4 ·1 ·· E31
D-4 01 ·· 305
··· ·· ·· 300
··· ·· ·· E30
```

Explanation: Glissando control `E31` changes note portamento behavior affecting commands `3xx`, `5xy` and `Mx`. Instead of stepless pitch bend (=glissando), the frequencies are rounded to nearest semitone. To revert to default glissando, use `E30`.

Notes: This command is not yet implemented in MilkyTracker.

E4x Vibrato control

Syntax:

E4

x = vibrato waveform selection

Example:

```
C-4 ·1 ·· 48C
··· ·· V0 E41
··· ·· V0 E42
··· ·· ·· E40
```

Explanation:

This command sets the waveform used for *4xy*, *6xy* and *Vx* vibrato commands. The default waveform is sine, reset on every new note (*E40*). Possible parameter *x* values are:

- *0* = Sine
- *1* = Ramp down
- *2* = Square
- *4* = Continuous sine
- *5* = Continuous ramp down
- *6* = Continuous square

Notes: This command is not yet implemented in MilkyTracker.

E5x Set note fine-tune

Syntax:

E5

x = fine-tune

Example:

```
C-4 ·1 ·· E54
··· ·· ·· ···
C-4 ·1 ·· E5C
··· ·· ·· ···
```

Explanation:

Sets note fine-tune overriding sample fine-tune setting. This command works a little differently for .MOD and .XM tracking. While both parameter value ranges are logical, the latter is also linear. See here:

x	ProTracker 2/3	Fasttracker II
0	0	-128
1	+16	-112
2	+32	-96
3	+48	-80
4	+64	-64
5	+80	-48
6	+96	-32
7	+112	-16
8	-128	0
9	-112	+16
A	-96	+32
B	-80	+48

C	-64	+64
D	-48	+80
E	-32	+96
F	-16	+112

E6x Pattern loop

Syntax:

E6

x = set loop point / number of iterations

Example:

```
C-4 ·1 ·· E60
··· ·· ·· ···
F-4 01 ·· ···
··· ·· ·· E63
```

Explanation: Loops a section of a pattern *x* times. *E60* sets the (optional) loop start point and *E6x* with *x* values *1–F* sets the end point and the number of iterations. If loop start point is not set, beginning of the pattern is used by default.

The loop points need to be set on the same channel for them to work correctly.

Notes:

Fasttracker II

One of the most (in)famous FT2 bugs is the *E60* bug: When *E60* is used on a pattern row *x*, the following pattern also starts from row *x* instead of the beginning of the pattern. This can be avoided by placing a *D00 pattern break* on the last row of the pattern where *E60* was used.

Tips:

Musicians concerned with correct playback of their .XM modules can utilize the *E60* bug to skip sections of (or the whole) song when played with lesser players. ;)

E7x Tremolo control

Syntax:

E7

x = tremolo waveform selection

Example:

```
C-4 ·1 ·· E72
··· ·· ·· 76C
··· ·· ·· E70
··· ·· ·· 700
```

Explanation: This command sets the waveform used for *7xy tremolo* command. As with *E4x vibrato control*, the default waveform is sine and the possible parameter *x* values are:

- *0* = Sine
- *1* = Ramp down
- *2* = Square
- *4* = Continuous sine
- *5* = Continuous ramp down
- *6* = Continuous square

E8x Set note panning position

Syntax: *E8*
x = panning position

Explanation: This command is another panning position command used by some trackers…

Notes: …However, since it does not work on Amiga (because of the hardware panning) nor in Fasttracker II (hmm, enough panning commands already?), effect relocation to *8xx* or *Px* is advised in order to produce compatible modules.

E9x Re-trigger note

Syntax: *E9*
x = triggering interval

Example:
```
C-4 ·1 ·· E93
C-4 ·1 ·· ···
··· ·· ·· ···
C-4 ·1 ·· ···
```

Explanation: This command re-triggers a note every *x* ticks.

EAx Fine volume slide up

Syntax: *EA*
x = speed

Example:
```
C-4 ·1 10 EA2
··· ·· ·· EA0
··· ·· ·· EA4
··· ·· ·· EA0
```

Explanation: Works similarly to *Ax0 volume slide* up, only the slide is a lot finer because the effect is applied only once per row.

EBx Fine volume slide down

Syntax: *EB*
x = speed

Example:
```
C-4 ·1 ·· EB2
··· ·· ·· EB0
··· ·· ·· EB4
··· ·· ·· EB0
```

Explanation: Works similarly to *A0y volume slide* down, only the slide is a lot finer like with *EAx*.

ECx Note cut

Syntax: `EC`
`x` = tick number

Example:
```
C-4 ·1 ·· EC1
C-4 ·1 ·· EC2
C-4 ·1 ·· ···
··· ·· ·· EC0
```

Explanation: Cuts a note by setting its volume to 0 at tick precision. Possible parameter `x` values are `0` – (song speed - 1). Higher values have no effect.

EDx Note delay

Syntax: `ED`
`x` = tick number

Example:
```
C-4 ·1 ·· ···
A#3 01 ·· ED3
C-4 ·1 ·· ···
··· ·· ·· ···
```

Explanation: Delays a note `x` ticks. Like with *`ECx` note cut*, possible `x` values are `0` – (song speed - 1). Higher values prevent the note from playing altogether.

EEx Pattern delay

Syntax: `EE`
`x` = amount of rows

Example:
```
C-4 ·1 ·· ···
A#3 01 ·· EE5
C-4 ·1 ·· ···
··· ·· ·· ···
```

Explanation: Delays playback progression for the duration of `x` rows.

Fxx Set song speed/BPM

Syntax: `F`
`xx` = speed/BPM value

Example:
```
C-4 ·1 ·· F90
A#3 01 ·· F03
C-4 ·1 ·· ···
··· ·· ·· ···
```

Explanation: Parameter `x` values `01` – `1F` set song speed i.e. the amount of ticks per row. Values `20` – `FF` set the BPM which essentially is the speed of the ticks. `F00` stops playback.

0 - Normal play or Arpeggio	0xy : x-first halfnote add, y-second
1 - Slide Up	1xx : upspeed
2 - Slide Down	2xx : downspeed
3 - Tone Portamento	3xx : up/down speed
4 - Vibrato	4xy : x-speed, y-depth
5 - Tone Portamento + Volume Slide	5xy : x-upspeed, y-downspeed
6 - Vibrato + Volume Slide	6xy : x-upspeed, y-downspeed
7 - Tremolo	7xy : x-speed, y-depth
9 - Set SampleOffset	9xx : offset (23 -> 2300)
A - VolumeSlide	Axy : x-upspeed, y-downspeed
B - Position Jump	Bxx : songposition
C - Set Volume	Cxx : volume, 00-40
D - Pattern Break	Dxx : break position in next patt
F - Set Speed	Fxx : speed (00-1F) / tempo (20-FF)
E9- Retrig Note	E9x : retrig from note + x vblanks

Other Exx commands:

E00/1=filter on/off - E1x/2x=FineSlide Up/Down - E30/1=tonep ctrl off/on
E40/1/2=Vib Waveform sine/rampdown/square, E70/1/2=Tremolo Waveform
E5x=set loop point,E6x=jump to loop+play x times
EAx/EBx=Fine volslide up/down
ECx/EDx=notecut after x vblanks/notedelay by x vblanks
EEx/EFx=PatternDelay by x notes/Invert loop, x=speed