SpessaSynth SFe 4.0

Sound bank format specification, an extension of SoundFont 2.04

Release Candidate

20 November 2026

This document is in the public domain. It is dedicated under CC0 1.0 Universal, see the LICENSE file of the repository.

Latest version: https://spessasus.github.io/spessasynth_sfe/

Contents

1. About This Document 4

2. Introduction 4

2.1 Scope and Intended Purpose of This Document 4

2.2 SFe Objectives 4

3. RIFF Structure 5

3.1 64-bit Chunk Sizes 5

4. SFe RIFF File Format 5

4.1 SFe RIFF File Format Level 0 5

4.2 SFe RIFF File Format Level 1 6

4.3 SFe RIFF File Format Level 2 6

4.4 SFe RIFF File Format Level 3 6

5. The INFO-list Chunk 7

5.1 The ifil Sub-chunk 7

5.2 The isng Sub-chunk 7

5.3 The XMOD Sub-chunk 7

5.4 The CURV Sub-chunk 8

5.5 The xdta-list Chunk 9

5.5.1 The Extended Modulator Records 10

6. The pdta-list Chunk 10

6.1 The PHDR Sub-chunk 10

6.2 The SHDR Sub-chunk 11

7. Enumerators 12

7.1 Generator and Modulator Destination Enumerators 12

7.1.1 Generator Enumerators Defined 14

7.1.2 Generator Summary 19

7.2 Modulator Source Enumerators 20

7.2.1 Custom Curves 21

7.3 Default Modulators 21

7.4 Ignored Modulators 23

8. Parameters and Synthesis Model 24

8.1 Synthesis Model 24

8.1.1 Wavetable Oscillator 25

8.1.2 Filter 25

8.1.3 Amplifier 26

8.1.4 Effects Sends 26

8.1.5 Low Frequency Oscillators 26

8.1.6 Envelope Generators 26

8.1.7 Modulation Interconnection Summary 27

9. Conversion Between Formats 27

9.1 Conversion to SoundFont 2 27

9.2 Conversion from SoundFont 2 30

10. Design Rationale 32

10.1 Default Modulators in Every File 32

10.2 Effects Send Amounts 32
10.3 Resonance 32
10.4 Envelope Time Controllers 33
10.5 Pan Amount 33
10.6 Velocity to Filter Cutoff 33
10.7 Brightness 33
10.8 Modulation Depth Range 34
10.9 Channel Pressure 34

11. MIDI Behavior 34
11.1 Modes 35
11.2 Preset Selection 35
11.2.1 Kinds of Preset 35
11.2.2 Channel State and Modes 36
11.2.3 Selecting a Preset 37
11.3 Controllers 38
11.4 Registered Parameters 39
11.5 Channel Mode and Pedal Controllers 40
11.6 System Exclusive Aliases 41
11.6.1 Master Parameters 41
11.6.2 GS Part Parameters 41
11.6.3 XG Part Parameters 42
11.6.4 Random Pan 43

SFe Format

> **Note**
>
> SpessaSynth SFe (with a lowercase "e") is not SFE (with an uppercase "E") by the SFE Team. In this document, SFe always means the SpessaSynth format, and SFE only means the SFE Team's format. This is not an official specification of SFE.
>
> The two formats share a name and many ideas, and SpessaSynth SFe is heavily inspired by SFE. They are still different formats, so a bank made for one is not a valid bank for the other.
>
> SpessaSynth SFe grew out of What SFe 4.0 should have been and SFe 5 Proposal #1: Initial Idea.

> **Status**
>
> This is a release candidate of SFe 4.0. It is published for review by implementers, and it may still change before the final 4.0. Feedback is welcome in the issue tracker.

1. About This Document

This document is the technical specification of SpessaSynth SFe (SF-Enhanced) version 4.0, a sound bank format that extends the SoundFont 2.04 format. It only describes what SFe adds or changes. Everything not described here is defined by the SoundFont 2.04 specification. A section that extends a part of SoundFont 2 names that part.

2. Introduction

2.1 Scope and Intended Purpose of This Document

This document is intended for developers of software that reads, writes or plays SFe files.

The terms BYTE, CHAR, WORD, SHORT, DWORD, FOURCC, chunk and sub-chunk have the same meaning as in SoundFont 2 sections 2 and 3. WORD and DWORD are unsigned. A QWORD is an unsigned 64-bit integer. All values are little-endian.

A reader is software that loads a file, a writer is software that stores one, and a synthesizer is software that plays one. "Must" marks a requirement on a reader, a writer or a synthesizer. "Should" marks a recommendation, and "may" marks something that is allowed.

2.2 SFe Objectives

SFe extends the SoundFont 2 synthesis model and does not add a second one. Every SFe feature:

1. Extends an existing SoundFont 2 stage (an envelope, the filter, an LFO, a modulator) instead of replacing it.
2. Defaults to the SoundFont 2 behavior. An SFe generator or chunk at its default leaves the behavior of SoundFont 2 alone. The rules that SFe changes for every file are listed in sections 7.1 to 7.4.
3. Is generally useful, and also lets a bank reproduce the behavior of classic General MIDI hardware.

SFe also includes the SoundFont 2 SpessaSynth extensions `DMOD` and `xdta`. This document defines them (sections 5.5, 7.3 and 9.1), and they are also described in soundfont-proposals.

An SFe file differs from a SoundFont 2 file only where this document says so. A bank without any SFe feature can always be written as a valid SoundFont 2 file (see section 9).

The MIDI behavior of a synthesizer (which messages it receives and what they do beyond the modulators of the bank) is outside this specification, apart from the part that decides how a bank sounds, which section 11 defines.

3. RIFF Structure

3.1 64-bit Chunk Sizes

This section extends the RIFF rules of SoundFont 2 section 3.

RIFF chunk sizes are DWORDs, which limits a file to 4 GB. An SFe file may use 64-bit chunk sizes instead. Such a file starts with the FOURCC `RIFS` in place of `RIFF`. In a `RIFS` file, every chunk (the RIFF chunk, all `LIST` chunks and all sub-chunks) has a 12-byte header: the FOURCC followed by a QWORD size.

All other RIFF rules are unchanged. The size does not include the header or the pad byte, a chunk of odd size is followed by one pad byte, and the data of a `LIST` chunk starts with its four-character list type.

A writer may use either form. A reader must detect the form from the first FOURCC of the file and must support both forms. A `RIFS` file must have the form type `sfen` (section 4.1), and a reader must reject a `RIFS` file with any other form type.

Only the chunk sizes grow. The sample addresses of the sample headers stay DWORDs (section 6.2), so the smpl sub-chunk still holds fewer than 2^32 sample data points. The addresses of a compressed sample are byte offsets, so a compressed sample must lie within the first 2^32 bytes of the smpl sub-chunk.

4. SFe RIFF File Format

4.1 SFe RIFF File Format Level 0

This section replaces SoundFont 2 section 4.1.

```
<SFe-form> -> RIFF ('sfen'              ; RIFS in a file with 64-bit chunk sizes
                {
                    <INFO-list>
                    <sdta-list>
                    <pdta-list>
                }
            )
```

The RIFF chunk of an SFe file has the form type `sfen` instead of `sfbk`. The form type differs on purpose, so that a SoundFont 2 reader rejects the file instead of playing it incorrectly.

4.2 SFe RIFF File Format Level 1

This section extends SoundFont 2 section 4.2.

```
<INFO-list> -> LIST ('INFO'
                {
                    <ifil-ck>           ; Version, major 4
                    <isng-ck>
                    <INAM-ck>
                    [<irom-ck>]
                    [<iver-ck>]
                    [<ICRD-ck>]
                    [<IENG-ck>]
                    [<IPRD-ck>]
                    [<ICOP-ck>]
                    [<ICMT-ck>]
                    [<ISFT-ck>]
                    <DMOD-ck>           ; Default modulators, see section 7.3
                    [<XMOD-ck>]         ; Extended records of the default modulators
                    [<CURV-ck>]         ; Custom curves
                    [<xdta-list>]       ; Extended hydra
                }
             )
```

The sub-chunks of the INFO-list chunk may appear in any order. A file that holds more than one `DMOD`, `XMOD` or `CURV` sub-chunk, or more than one `xdta-list` chunk, is not valid, and a reader must reject it.

The `sdta-list` and `pdta-list` chunks and their sub-chunks have the same layout as in SoundFont 2. Sections 5.5.1, 6.1 and 6.2 change the meaning of some of their fields.

4.3 SFe RIFF File Format Level 2

This section extends SoundFont 2 section 4.3.

```
<DMOD-ck> -> DMOD(<pmod-rec>)           ; The default modulators, like pmod
<XMOD-ck> -> XMOD(<xmod-rec>)           ; The extended records of DMOD
<CURV-ck> -> CURV(dwCurveCount:DWORD     ; The number of curves
                  <curv-rec>...)        ; dwCurveCount curves
<xdta-list> -> LIST ('xdta'
                {
                    <phdr-ck>           ; The nine sub-chunks of pdta, in the same order
                    <pbag-ck>
                    <pmod-ck>           ; Holds <xmod-rec> records
                    <pgen-ck>
                    <inst-ck>
                    <ibag-ck>
                    <imod-ck>           ; Holds <xmod-rec> records
                    <igen-ck>
                    <shdr-ck>
                }
             )
```

4.4 SFe RIFF File Format Level 3

This section extends SoundFont 2 section 4.4.

```
<xmod-rec> -> struct sfModExtension
              {
                  DWORD dwSourceCurve;
                  DWORD dwAmountCurve;
                  BYTE abyReserved[8];
              };

<curv-rec> -> struct sfCurve
              {
                  CHAR achCurveName[40];
                  WORD wPointCount;
                  struct sfCurvePoint
                  {
                      WORD x;
                      WORD y;
                  } points[wPointCount];
              };
```

5. The INFO-list Chunk

5.1 The ifil Sub-chunk

This section extends SoundFont 2 section 5.1.

An SFe file is a file with the form type `sfen` and an `ifil` `wMajor` of 4. This document defines version 4.0, so a writer must write `wMajor` 4 and `wMinor` 0. Later minor versions may add generators, source types, transform types, modulator links (section 7.4) and sub-chunks of the INFO-list chunk. New sub-chunks only ever appear in the INFO-list chunk, as SoundFont 2 section 10.2 allows. A reader must load a file with a higher `wMinor` than it knows, and ignore what it does not understand, as SoundFont 2 asks: the generators, the source types and the sub-chunks of the INFO-list chunk, and every modulator that uses a transform type, a source or a destination it does not know (section 7.4). An ignored modulator supersedes nothing. Unknown chunks anywhere else follow the rules of SoundFont 2. A `wMajor` above 4 is not defined by this document, and a reader should reject such a file.

A file with the form type `sfbk` is a SoundFont 2 or SoundFont 3 file whatever its `ifil` says, and it is never read as an SFe file.

A file with the form type `sfen` and a `wMajor` below 4 was written for an earlier, incompatible draft of SFe. It is not an SFe file. A reader that does not reject it reads it as a SoundFont 2 file, which means that it ignores all generators and modulator destinations above 60 and loads the file in one of the two modes of section 9.2.

5.2 The isng Sub-chunk

This section extends SoundFont 2 section 5.2.

A bank made for this specification should hold the string "SFe 4" in its `isng` sub-chunk.

5.3 The XMOD Sub-chunk

The XMOD sub-chunk holds the extended records of the default modulators in `DMOD`. Each record is an `sfModExtension` structure, 16 bytes in length:

```
struct sfModExtension
{
    DWORD dwSourceCurve;
    DWORD dwAmountCurve;
    BYTE abyReserved[8];
};
```

The sub-chunk contains one record for each modulator of `DMOD` that has the extended flag (section 5.5.1), in the same order, followed by a terminal record with zero in all fields. The fields are defined in section 5.5.1, and the records are matched to the modulators as that section describes.

The sub-chunk is optional. A writer must write it when a modulator of `DMOD` has the flag, which every one that uses a custom curve must have. If the XMOD sub-chunk is missing, if its size is not a multiple of 16 bytes, or if its record count does not match, every default modulator that has the flag must be ignored. `DMOD` itself keeps the layout of `pmod`, so readers that do not know XMOD keep reading it.

5.4 The CURV Sub-chunk

The CURV sub-chunk is an optional sub-chunk holding the custom curves of the bank (section 7.2.1). It starts with the DWORD `dwCurveCount`, the number of curves, followed by that many `sfCurve` structures of variable length:

```
struct sfCurvePoint
{
    WORD x;
    WORD y;
};

struct sfCurve
{
    CHAR achCurveName[40];
    WORD wPointCount;
    sfCurvePoint points[wPointCount];
};
```

Curves are numbered from 0 in the order they are stored. A curve may be stored even if no modulator uses it, and a reader must keep it.

The ASCII character field `achCurveName` contains the name of the curve, terminated by a zero valued byte unless it fills all 40 bytes. Curve names need not be unique and may be empty.

The WORD `wPointCount` must be at least 2. The `x` values range from 0 to 16384 and must never decrease from one point to the next. The `y` values range from 0 to 16384.

A source value `x` reaches 16384 at full scale (section 7.2), so the `x` of 16384 is the end of the range. A point there lets a curve be defined up to exactly 1: the points (0, 0) and (16384, 16384) make the linear curve, where a source value `x` outputs `x / 16384`. The reversed direction reaches 16384 too (section 7.2.1), because a source value of 0 becomes 16384.

A curve that breaks these rules is invalid, and it keeps its number. Every modulator that refers to an invalid curve, or to a curve number of `dwCurveCount` or higher, must be ignored.

The size of the sub-chunk must be exactly 4 bytes for `dwCurveCount` plus the size of every curve it counts. A sub-chunk that ends before all `dwCurveCount` curves are read, or that holds bytes after the last one, is malformed. An SFe file with a malformed `CURV` sub-chunk is not valid, and a reader must reject it.

5.5 The xdta-list Chunk

The optional `xdta-list` chunk is a `LIST` chunk inside the INFO-list chunk. It is defined by the SoundFont 2 extended limits proposal, and SFe keeps its layout and gives its `pmod` and `imod` records a meaning (section 5.5.1). A reader of SFe files must read it unless it is structurally unsound (below), and so must a reader that loads a SoundFont 2 file in the extended compatibility mode (section 9.2). It lifts these SoundFont 2 limits:

- Names of presets, instruments and samples may be up to 40 characters long.
- The bag indices of the preset and instrument headers, the generator and modulator indices of the bags, and `wSampleLink` are 32 bits wide.

The `instrument` and `sampleID` generators stay 16 bits wide. A zone refers to an instrument or a sample beyond the first 65536 with the generators `instrumentMSB` and `sampleIDMSB` (section 7.1.1). SoundFont 2 has no such generators, so a writer of a SoundFont 2 file must not write a bank in which a zone refers beyond them. `wSampleLink` is widened all the same, so that an editor can link any two samples.

The chunk must contain nine sub-chunks in this order, each with the same record size as its counterpart in `pdta`. The exception is `pmod` and `imod` in an SFe file, whose records are 16 bytes (section 5.5.1). `phdr`, `pbag`, `inst`, `ibag` and `shdr` hold one record for each record of their counterpart, including the terminal record. `pmod` and `imod` hold one record for each modulator with the flag, plus the terminal record (section 5.5.1), and `pgen` and `igen` hold only the terminal record:

Sub-chunk	Fields used
`phdr`	`achPresetName`: characters 21 to 40 of the name. `wPresetBagNdx`: the top 16 bits.
`pbag`	`wGenNdx` and `wModNdx`: the top 16 bits.
`pmod`	The extended modulator records (below).
`pgen`	Only the terminal record.
`inst`	`achInstName`: characters 21 to 40 of the name. `wInstBagNdx`: the top 16 bits.
`ibag`	`wInstGenNdx` and `wInstModNdx`: the top 16 bits.
`imod`	The extended modulator records (below).
`igen`	Only the terminal record.
`shdr`	`achSampleName`: characters 21 to 40 of the name. `wSampleLink`: the top 16 bits.

All other fields must be written as 0, and a reader must ignore them. A name continues in `xdta` only when it is longer than 20 characters. A reader takes the `pdta` half up to its first zero byte, or all 20 characters when it has none. Only in the second case it appends the `xdta` half, up to its first zero byte or all 20 characters when it has none. A `pdta` half with a zero byte ends the name, and the `xdta` half is then ignored. An index is `(top << 16) | bottom`, with the bottom 16 bits in `pdta`.

When the `xdta-list` chunk is used (it is not ignored), every rule of SoundFont 2 about these indices applies to the combined index. This includes the checks of the terminal records against the sizes of the bag, generator and modulator sub-chunks, so a reader must not check the 16-bit values of `pdta` alone.

If the nine sub-chunks are not all present in this order, or if `phdr`, `pbag`, `inst`, `ibag` or `shdr` differs in size from its `pdta` counterpart, the `xdta-list` chunk is structurally unsound and the whole chunk must be ignored. A reader must ignore sub-chunks after `shdr` and the contents of `pgen` and `igen`.

5.5.1 The Extended Modulator Records

A modulator has an extended record when bit 15 (0x8000) of its transform enumerator (`sfModTransOper`) is set. This bit is the extended flag. In an SFe file the transform itself is the lower 15 bits. Only some modulators need a record, so the records are sparse.

A modulator that uses a custom curve must have the flag. It uses a custom curve when its source or its amount source has the source type 4 (section 7.2.1). A modulator that uses a custom curve without the flag must be ignored.

The `pmod` and `imod` sub-chunks of `xdta` hold one `sfModExtension` record (section 5.3) for each modulator of their `pdta` counterpart that has the flag, followed by a terminal record with zero in all fields. A modulator without the flag has no record. A writer must write the `xdta-list` chunk when a modulator of `pmod` or `imod` has the flag.

A reader matches them in order. It walks the modulators of the `pdta` sub-chunk in order, without the terminal record, and every modulator with the flag takes the next record. Every such modulator counts, including one the reader ignores for another reason (an unknown destination, for example), so all readers match the records the same way. The number of records without the terminal must equal the number of modulators with the flag. If it does not, or if the size of the sub-chunk is not a multiple of 16 bytes, the records cannot be matched, and every modulator of that sub-chunk that has the flag must be ignored.

- The DWORD `dwSourceCurve` is the index into the CURV sub-chunk of the curve used by the modulator's source, when the source's type is 4 (section 7.2.1). Otherwise, it must be 0.
- The DWORD `dwAmountCurve` is the same for the modulator's amount source.
- A reader must ignore an index whose source is not of type 4. When the other source is of type 4, the modulator is kept.
- The 8 bytes of `abyReserved` are reserved for later versions. Writers must write 0. A reader that finds a non-zero byte must ignore that modulator, since a later version may have changed what it does. It still takes the record.
- A reader that does not understand why a modulator has a record must ignore the modulator, and it still takes the record. In version 4.0 the only reason is a custom curve, so a modulator with the flag and no source of type 4 is ignored. A later minor version may give records to other modulators in the same way, and the records of the other modulators still match.

If the `xdta-list` chunk is missing or ignored, the modulators of `pmod` and `imod` have no extended records, and every one of them that has the flag or uses a custom curve must be ignored. SoundFont 2 files with extended limits hold only a terminal record of the size of a `pdta` modulator record, which has no extended records to match.

The XMOD and CURV sub-chunks, the extended records and the flag must be ignored in a file that is not an SFe file. In such a file bit 15 is part of the transform, which then is not a known one (section 7.4).

6. The pdta-list Chunk

6.1 The PHDR Sub-chunk

This section extends SoundFont 2 section 7.2.

In an SFe file, the WORD `wBank` of each preset header holds three values:

Bits	Value
0-6	bank MSB (MIDI CC 0)
7	the drum flag
8-14	bank LSB (MIDI CC 32)
15	reserved

A preset with the drum flag is a percussion preset, selected for a percussion channel as a preset at bank 128 is in SoundFont 2. The drum flag may be combined with any bank MSB and LSB, and section 11.2 defines how such a preset is selected. Bit 15 must be written as 0 and must be ignored when read.

For example, a preset at bank MSB 0 and LSB 66 is stored as 0x4200, and a drum preset at bank MSB 0 and LSB 0 is stored as 0x0080.

This layout is the rule for SFe files. A SoundFont 2 `wBank` from 0 to 128 decodes to the same bank with this layout, and a SoundFont 2 file that is loaded in the extended compatibility mode (section 9.2) must be decoded this way, so that its presets have the four values of section 11.2.1. The SoundFont 2 rule that a preset with a `wBank` above 128 cannot be played applies to neither.

6.2 The SHDR Sub-chunk

This section extends SoundFont 2 section 7.10.

A sample may be compressed, in an SFe file and in a SoundFont 3 file alike (an `sfbk` file with `ifil` 3.0, section 9.1). A compressed sample has bit 4 (0x10) set in its `sfSampleType`. Its data in the smpl sub-chunk is an Ogg Vorbis stream:

- `dwStart` is the byte offset of the first byte of the stream in the smpl sub-chunk, and `dwEnd` is the byte offset just past its last byte.
- `dwStartloop` and `dwEndloop` are sample data points relative to the start of the decoded sample.
- The sample ends after its last decoded sample data point. This is the end that the address offset generators and section 8.1.1 refer to.
- The stream must hold one channel at the sample rate `dwSampleRate`. When it does not, a reader takes the decoded sample data points of the first channel as they are, without resampling, and plays them at `dwSampleRate`.
- A stream that cannot be decoded plays as silence, and the file stays valid.

In an SFe file, and in a file loaded in the extended compatibility mode (section 9.2), a reader must decide per sample from this flag and must not use `ifil`.

The 46 zero valued sample data points that SoundFont 2 asks for after each sample are not stored after a compressed sample. A writer must follow a stream of odd length with one pad byte of 0, so that the next sample starts at an even byte offset, as an uncompressed sample data point takes 2 bytes.

The `sm24` sub-chunk keeps its SoundFont 2 layout: its byte `n` belongs to the 16-bit sample data point at byte `2 * n` of the smpl sub-chunk. The bytes that belong to compressed data must be written as 0 and are ignored, so only the uncompressed samples have 24 bits.

SFe 4.0 defines Ogg Vorbis as the only compression. A writer must not store sample data compressed in any other way.

The four sample addresses stay DWORDs, and `xdta` does not extend them.

A synthesizer must not use the sample links. Every zone plays by itself, with its own generators, and `wSampleLink` and the left, right and linked bits of `sfSampleType` only inform an editor. The two zones of a stereo pair are not tied together, so the pitch of the left sample does not follow the generators of the right one.

7. Enumerators

7.1 Generator and Modulator Destination Enumerators

This section extends SoundFont 2 section 8.1.

SoundFont 2 defines generators up to 60 (`endOper`). SFe defines the generators 61 to 89. They must be handled as SoundFont 2 value generators, and they are valid modulator destinations. The generators 90 and 91 extend `instrument` and `sampleID` instead (below). In a file that is not an SFe file, a reader must ignore the generators 61 to 91, as SoundFont 2 asks for unknown generators.

The generators `filterType`, `modLfoWaveform`, `vibLfoWaveform` and `ignoreNoteOff` are only valid at the instrument level and must be ignored in preset zones. A modulator in a preset zone whose destination is only valid at the instrument level must be ignored too. This covers the four generators above and the SoundFont 2 generators that are only valid at the instrument level.

The four generators are enumerations. A stored value outside the defined values must be treated as the default, so that a reader of this version plays a value that a later version defines as the default and not as a neighbor. The default replaces such a stored value before the modulators are added. When a modulator targets one of the four, the value after the modulators is clamped to the generator's range and truncated toward zero. Like most other generators, the four follow their modulators during the note.

Some SoundFont 2 generators do not take modulators like the others:

- The address offsets (`startAddrsOffset`, `endAddrsOffset`, `startloopAddrsOffset`, `endloopAddrsOffset` and their coarse forms) take their modulators once, when the voice starts. A change during the note has no effect. Each of these generators is truncated toward zero after its modulators are added, before a coarse value is multiplied by 32768.
- `overridingRootKey`, `exclusiveClass` and `sampleModes` are plain values. A synthesizer should ignore a modulator whose destination is one of them. It may apply such modulators, reading them once when the voice starts and truncating the value toward zero, but this is not recommended. A bank should not rely on them.
- A modulator whose destination is `keyRange`, `velRange`, `keynum`, `velocity`, `instrument`, `sampleID`, `instrumentMSB` or `sampleIDMSB` must be ignored. These generators select the zone or set up the voice before its modulators are evaluated.

`scaleTuning`, which SoundFont 2 also marks as non-real-time, follows its modulators during the note like the other generators.

In SFe files, the range of two SoundFont 2 generators is wider. `exclusiveClass` ranges from 0 to 32767, where SoundFont 2 stops at 127, and `fineTune` ranges from -12700 to 12700 cents, where SoundFont 2 stops at -99 and 99.

The scope of `exclusiveClass` is the MIDI channel, where SoundFont 2 names the preset. A voice with a class other than 0 ends every voice of the same class that sounds on its channel, also one that was started before a Program Change. Voices started by the same note on do not end each other. The ended

voice should get a very short release and should not stop at once.

Generator sums saturate. The instrument value plus the preset value must be limited to -32768 to 32767. The outputs of all modulators of the generator are then added to that value without a limit in between, and only the final sum is limited to -32768 to 32767, so the order of the modulators never matters. The limits of a generator are applied once, to that final sum. For the delay, attack, hold and fade times below, the lower limit is -32768, so the zero time rule sees the final sum.

In SFe files, all delay, attack, hold and fade times follow the convention SoundFont 2 gives for its delay, attack and hold times, where -32768 means zero time. It applies to the SoundFont 2 generators `delayModLFO`, `delayVibLFO`, `delayModEnv`, `attackModEnv`, `holdModEnv`, `delayVolEnv`, `attackVolEnv` and `holdVolEnv`, and to the SFe generators `fadeModLFO`, `fadeVibLFO`, `delayModEnv2`, `attackModEnv2`, `holdModEnv2` and `delayVoice`:

- The value is taken after the preset offset and the modulators, and before key scaling (`keynumToModEnvHold`, `keyNumToModEnv2Hold` and `keynumToVolEnvHold`).
- A value of -32768 means exactly zero time.
- A value from -32767 to -12001 must be treated as -12000, as in the SoundFont 2 table. As a result, a negative preset offset or modulator only makes these times shorter than 1 ms when the sum reaches -32768.
- The defaults of the SoundFont 2 generators do not change.

A synthesizer may round a time other than zero time that is shorter than the period at which it updates its envelopes and LFOs.

> **Important**
>
> This is a rule for SFe files. An SFe synthesizer must play -32768 as zero time in an SFe file, and not as 1 ms. For a SoundFont 2 file it applies only in the extended compatibility mode (section 9.2).

When `delayVolEnv` and `attackVolEnv` are both zero time (-32768), the voice must start at its peak level, without a fade in. A value of -12000 is 1 ms and not zero time.

In SFe files, the "100% change" that times the decay and the release of the Volume Envelope is 960 cB, where SoundFont 2 names 100 dB. A `sustainVolEnv` above 960 counts as 960, for the sustain level and for the decay toward it.

In SFe files, `sampleModes` is an enumeration of the whole value, where SoundFont 2 reads its two lowest bits as flags:

- **0, 1 and 3** are as in SoundFont 2.
- **2, start on release.** SoundFont 2 reads this value as no loop. The voice is silent until it is released, and its envelopes and LFOs run from the note on. When the voice is released (at the note off, or later when the sustain pedal or All Notes Off releases it), the sample starts from its start point (`dwStart` plus its offsets) and plays once, without looping. Its Volume Envelope is then in its release, so the release time limits how much of the sample is heard. With `ignoreNoteOff` at 1 when the note off arrives, the sample never starts and the voice ends at that note off.
- **4, alternating loop.** The playback position moves forward from the loop start point (`dwStartloop` plus its offsets) to the sample data point before the loop end (`dwEndloop` minus 1, plus its offsets), then backward to the loop start point, and repeats. Each turn point is played once, so the loop takes `2 * (end - start - 1)` sample data points per round trip, where `end` is exclusive as in SoundFont

2. The sample data points are played as stored in both directions. The voice loops through the release phase too, like the value 1. A loop of a single sample data point plays like the value 1.

- **Any other value** must be treated as 0.

The `sampleModes` value is read once, when the voice starts.

Past a turn point, interpolation must use the sample data points the playback position reaches next, not the ones stored beyond the loop.

> **Note**
>
> The value 2 was introduced by Polyphone.

The `initialAttenuation` generator is in true centibels, the same unit as `gain` and as the amounts of the modulators that target it. An SFe synthesizer does not scale it. SoundFont 2 names centibels too, but SoundFont 2 banks follow the EMU8000, which plays the value at 0.4 times its size, so they store it 2.5 times as large. Section 9 converts it when a SoundFont 2 file is read in the extended compatibility mode or written. The limits of the generator with its modulators are 0 to 1440 cB. SoundFont 2 limits the value to 1440 in its own unit, which is 576 cB, so a converted bank stays inside the range and a bank made for SFe can use all of it.

7.1.1 Generator Enumerators Defined

61 filterType

The response of the voice filter.

- 0 is the lowpass filter of SoundFont 2. SoundFont 2 only idealizes it, and SFe defines it as the second order response below.
- 1 is a highpass filter.
- 2 is a bandpass filter.
- 3 is a peak filter.
- 4 is a notch filter.

All other filter parameters drive every type in the same way: `initialFilterFc`, `initialFilterQ`, `filterQOffset`, `maxFilterFc`, their modulators, the LFOs and the Modulation Envelopes.

The responses are defined as follows, with `s` the Laplace variable, `w` the cutoff, `Q` the linear Q and `D = s^2 + w * s / Q + w^2`:

Type	Response	Notes
lowpass	`w^2 / D`	The SoundFont 2 filter.
highpass	`s^2 / D`	
bandpass	`w * s / D`	Gain Q at the cutoff.
peak	`(w^2 - s^2) / D`	Gain 2 * Q at the cutoff and 1 at both ends.
notch	`(w^2 + s^2) / D`	A zero at the cutoff.

A synthesizer may realize these responses in any way that approximates them closely. The bilinear transform with a prewarped cutoff is recommended and is close enough. The filter should stay stable at every resonance this section allows, and a synthesizer may limit Q where it cannot compute a higher one stably.

The linear Q is `10^((r - 30.1) / 200)`, with `r` the resonance in centibels (see `filterQOffset`). SFe defines Q this way, so an `r` of 0 gives a Butterworth response with no peak. It puts the lowpass peak about 3 dB below the literal wording of SoundFont 2.

The response is scaled by `10^(-q / 400)`, the SoundFont 2 DC gain reduction of half the resonance (`q / 20` dB), with `q` the part of `r` that comes from `initialFilterQ` (see `filterQOffset`). `filterQOffset` does not change the gain. The scaling applies to every type and scales the whole response. For example, the ends of the peak and notch filters sit at `-q / 20` dB.

For every type, `initialFilterFc` plus its modulators is clamped to 1500 to 13500 absolute cents, as in SoundFont 2. The excursions of the LFOs and the Modulation Envelopes are added after that clamp. The final cutoff has no lower limit. It should be limited to 0.45 times the sample rate of the synthesizer, since a digital filter cannot follow the response above that.

SoundFont 2 states that the filter does not affect the signal when the cutoff exceeds 20 kHz and the Q is zero. So a synthesizer may skip the lowpass filter when the cutoff after the ceiling of `maxFilterFc` (section 8.1.2, step 3), before the limit for the sample rate, is at 13500 absolute cents or above and `r` is 0. The gain reduction of `q` still applies then. The other types must always be computed, because at 13500 absolute cents a highpass or bandpass filter is not neutral.

62 maxFilterFc

The highest cutoff the voice filter can reach, in absolute cents.

The ceiling applies to the final cutoff (`initialFilterFc` plus its modulators, the LFOs and the Modulation Envelopes), before the filter is computed. A value of 13500, the default, is defined as "no ceiling". The final cutoff then has no limit other than the one for the sample rate (see `filterType`).

For example, with a ceiling of 9000 neither the velocity nor an envelope can open the filter past about 1.5 kHz.

63 filterQOffset

An offset, in centibels, added to the resonance of the voice filter. The resonance is:

```
q = clamp(initialFilterQ + its modulators, 0, 960)
r = q + clamp(filterQOffset + its modulators, -960, 960)
```

This generator allows for a Q lower than 0.707 (a negative `r`) and a resonance above the SoundFont 2 maximum. It applies to every filter type, and the sum is not clamped further. Unlike `initialFilterQ`, it does not change the gain of the filter: the DC gain reduction of section 7.1.1 is `q / 20` dB, whatever `r` is. So it moves the resonance without lowering the level.

> **Note**
>
> For a negative `r` the response falls off from about `w * Q`, so at -300 it starts more than 5 octaves below the cutoff. Above 960 the filter rings for a very long time.

64 vibLfoToFilterFc

The degree, in cents, to which a full scale excursion of the Vibrato LFO influences the filter cutoff. It works exactly like `modLfoToFilterFc` does for the Modulation LFO.

65 vibLfoToVolume

The degree, in centibels, to which a full scale excursion of the Vibrato LFO influences the volume. It works exactly like `modLfoToVolume` does for the Modulation LFO.

66 modLfoToAmplitude

The degree, in 0.1 % units, to which the Modulation LFO changes the voice amplitude linearly.

The two linear depths (66 and 67) are summed before one clamp. The voice output is multiplied by:

```
clamp(1 + lfo_mod * modLfoToAmplitude / 1000 + lfo_vib * vibLfoToAmplitude / 1000, 0, 2)
```

where `lfo_mod` and `lfo_vib` are the LFO outputs from -1 to 1 after the LFO delay and fade. A depth of 1000 swings the voice from silence to twice its amplitude, and a negative depth starts by turning the voice down.

The factor applies together with the Volume Envelope, after the filter. It is separate from the centibel depths (`modLfoToVolume`, `vibLfoToVolume`).

67 vibLfoToAmplitude

The same as `modLfoToAmplitude`, for the Vibrato LFO.

68 fadeModLFO

The time, in absolute timecents, over which the output of the Modulation LFO ramps linearly from zero to full after its delay (`delayModLFO`) ends. It scales all the depths of that LFO. The default of -32768 is zero time, so there is no fade.

69 fadeVibLFO

The same as `fadeModLFO`, for the Vibrato LFO and `delayVibLFO`.

70 modLfoWaveform

The waveform of the Modulation LFO.

With `p` the position in the cycle, from 0 up to but not including 1, the output is:

Value	Waveform	Output
0	The SoundFont 2 triangle	`4p` for `p < 0.25`, `2 - 4p` for `p < 0.75`, `4p - 4` from there on.
1	A rising sawtooth	`2p` for `p < 0.5`, `2p - 2` from there on.
2	A sine	`sin(2 * pi * p)`
3	A square wave	1 for `p < 0.5`, -1 from there on.

Every waveform is at zero and rising at `p = 0`, except the square, which begins its half at 1. The LFO starts at the position given by the start phase (`modLfoStartPhase`).

71 vibLfoWaveform

The same as `modLfoWaveform`, for the Vibrato LFO.

72 modLfoStartPhase

The position in its cycle, in degrees, where the Modulation LFO starts: `p = modLfoStartPhase / 360`, with `p` as in `modLfoWaveform`. A value of 0 is the SoundFont 2 start, and 360 is the same as 0. For the triangle and the sine, 90 starts at the peak, 180 at zero and falling, and 270 at the trough.

The value is read once, when the voice starts. The preset offset and the modulators add to it like to any other generator, and the sum is clamped to 0 to 360, not wrapped. An instrument value of 270 plus a preset value of 180 gives 360, which starts like 0.

73 vibLfoStartPhase

The same as `modLfoStartPhase`, for the Vibrato LFO.

74 modEnv2ToPitch

The degree, in cents, to which a full scale excursion of the second Modulation Envelope influences pitch. It works like `modEnvToPitch`.

75 modEnv2ToFilterFc

The degree, in cents, to which a full scale excursion of the second Modulation Envelope influences the filter cutoff. It works like `modEnvToFilterFc`.

76 to 83 The Second Modulation Envelope

The parameters of the second Modulation Envelope. Each is defined like the generator of the first Modulation Envelope in the same row:

#	Generator	SoundFont 2 equivalent
76	`delayModEnv2`	`delayModEnv`
77	`attackModEnv2`	`attackModEnv`
78	`holdModEnv2`	`holdModEnv`
79	`decayModEnv2`	`decayModEnv`
80	`sustainModEnv2`	`sustainModEnv`
81	`releaseModEnv2`	`releaseModEnv`
82	`keyNumToModEnv2Hold`	`keynumToModEnvHold`
83	`keyNumToModEnv2Decay`	`keynumToModEnvDecay`

Only the defaults differ from the first envelope (see section 7.1.2). The lower limit of -32768 of the delay, attack and hold times applies to both envelopes (section 7.1).

The second envelope is independent of the first, so a zone can have separate pitch and filter envelopes. With `modEnv2ToPitch` and `modEnv2ToFilterFc` at 0 (the default) it has no effect and its computation may be skipped.

84 decay2VolEnv

The time, in absolute timecents, of a second decay segment of the Volume Envelope. It is measured like `decayVolEnv`, as the time for a change of 960 cB (section 7.1).

After the decay reaches `sustainVolEnv`, the level continues in a straight line in centibels to the end level, `sustainVolEnv + sustain2VolEnv`, which is clamped to 0 to 960 cB. A `sustainVolEnv` above 960 counts as 960 (section 7.1), both as the start of the segment and in this sum.

As with the decay, shorter distances take proportionally less time, and `keynumToVolEnvDecay` scales this time too. A release during the segment starts from the current level.

85 sustain2VolEnv

The end of the second decay segment, in centibels relative to `sustainVolEnv`. A negative value makes the level rise back up. The default of 0 means there is no second segment.

86 gain

This is the gain, in centibels, added to the level of the voice. A positive value is louder. The total attenuation of the voice is:

```
clamp(initialAttenuation + its modulators, 0, 1440) - clamp(gain + its modulators, -1440, 1440)
```

The result is not clamped at 0, so a voice can play above unity. `gain` and `initialAttenuation` are in the same unit, so a `gain` of 100 cancels an `initialAttenuation` of 100.

87 delayVoice

The time, in absolute timecents, by which the start of the whole voice is delayed after its note on: the sample, all three envelopes and both LFOs. The voice is silent before that. A synthesizer should start the voice as close to that time as it can. The value is read once, at the note on.

A note off before the voice starts does not cancel it. The voice still starts at its time and then releases at once, as if the note off arrived when it started. The voice takes part in its exclusive class from the note on, not from its delayed start. A voice that its exclusive class ends before it starts never sounds, and a minimum note length of section 8.1 does not apply to it. The default of -32768 is zero time, so there is no delay.

88 ignoreNoteOff

If 1, a note off does not release the voice and the sustain pedal does not hold it. The voice plays until its Volume Envelope or its sample ends, even for very short notes, except with `sampleModes` 2 (section 7.1). The value is read when the note off arrives.

All Notes Off (CC 123) still releases the voice. Another note in the same exclusive class and All Sound Off (CC 120) end it (sections 7.1 and 11.5). A system reset ends or releases it (section 11.1).

For example, a drum zone can then play its whole Volume Envelope however short the note is.

89 variationEffectsSend

The degree, in 0.1 % units, to which the audio output of the voice is sent to the variation effects processor. It works like `reverbEffectsSend` and `chorusEffectsSend` do for their processors.

Support for this generator is optional. If the synthesizer has an additional effects processor beyond reverb and chorus, such as a delay or a user-selectable effect, this generator sets how much of the voice is sent to it. A synthesizer without such a processor must ignore this generator and the modulators that target it.

90 instrumentMSB and 91 sampleIDMSB

The upper 16 bits of the index of `instrument` and of `sampleID`, so that a zone can refer to an instrument or a sample beyond the first 65536. The index is `(instrumentMSB << 16) | instrument` in a preset zone and `(sampleIDMSB << 16) | sampleID` in an instrument zone. All four are read as unsigned WORDs, and the default of both MSB generators is 0.

`instrumentMSB` only counts in a preset zone that has `instrument`, and `sampleIDMSB` only in an instrument zone that has `sampleID`. Each must come immediately before the generator it extends, which stays the last generator of the zone. Anywhere else (a global zone, the other level, or after the generator it extends) it is ignored, and the zone itself is not: as in SoundFont 2, generators after `instrument` or `sampleID` are ignored. They are discarded first, so `instrument` or `sampleID` counts as the last generator

of the zone (SoundFont 2 sections 7.4 and 7.9). When a zone holds two of them, only the one immediately before counts. Modulators cannot target them (section 7.1). A writer writes one only when the index is above 65535.

The index of `instrument` and `sampleID`, with or without these generators, must refer to an instrument or a sample of the bank, not to its terminal record. A reader must ignore a zone whose index refers past them, where SoundFont 2 rejects the file.

7.1.2 Generator Summary

#	Name	Unit	Min	Min, real units	Max	Max, real units	Default
61	`filterType` +	type	0	lowpass	4	notch	0 (lowpass)
62	`maxFilterFc`	absolute cents	1500	20 Hz	13500	20 kHz	13500 (none)
63	`filterQOffset`	cB	-960	-960 (Q 0.00001)	960	960 (Q 44600)	0
64	`vibLfoToFilterFc`	cents fs	-12000	-10 octaves	12000	10 octaves	0
65	`vibLfoToVolume`	cB fs	-960	-96 dB	960	96 dB	0
66	`modLfoToAmplitude`	0.1 % fs	-1000	-100 %	1000	100 %	0
67	`vibLfoToAmplitude`	0.1 % fs	-1000	-100 %	1000	100 %	0
68	`fadeModLFO`	timecents	-32768	0 s	5000	18 s	-32768 (no fade)
69	`fadeVibLFO`	timecents	-32768	0 s	5000	18 s	-32768 (no fade)
70	`modLfoWaveform` +	waveform	0	triangle	3	square	0 (triangle)
71	`vibLfoWaveform` +	waveform	0	triangle	3	square	0 (triangle)
72	`modLfoStartPhase` @	degrees	0	0 degrees	360	360 degrees	0
73	`vibLfoStartPhase` @	degrees	0	0 degrees	360	360 degrees	0
74	`modEnv2ToPitch`	cents fs	-12000	-10 octaves	12000	10 octaves	0
75	`modEnv2ToFilterFc`	cents fs	-12000	-10 octaves	12000	10 octaves	0
76	`delayModEnv2`	timecents	-32768	0 s	5000	18 s	-32768 (no delay)
77	`attackModEnv2`	timecents	-32768	0 s	8000	101.6 s	-32768 (instant)
78	`holdModEnv2`	timecents	-32768	0 s	5000	18 s	-32768 (no hold)
79	`decayModEnv2`	timecents	-12000	1 ms	8000	101.6 s	-12000 (1 ms)
80	`sustainModEnv2`	0.1 % decrease	0	100 %	1000	0 %	0 (100 %)
81	`releaseModEnv2`	timecents	-12000	1 ms	8000	101.6 s	-12000 (1 ms)
82	`keyNumToModEnv2Hold`	timecents per key	-1200	doubles per key	1200	halves per key	0
83	`keyNumToModEnv2Decay`	timecents per key	-1200	doubles per key	1200	halves per key	0
84	`decay2VolEnv`	timecents	-12000	1 ms	8000	101.6 s	-12000 (1 ms)
85	`sustain2VolEnv`	cB relative to the sustain	-1440	-960 (96 dB up)	1440	960 (96 dB down)	0 (none)
86	`gain`	cB	-1440	-144 dB	1440	144 dB	0
87	`delayVoice` @	timecents	-32768	0 s	5000	18 s	-32768 (no delay)

#	Name	Unit	Min	Min, real units	Max	Max, real units	Default
88	`ignoreNoteOff` +	switch	0	off	1	on	0 (off)
89	`variationEffectsSend`	0.1 %	0	0 %	1000	100 %	0
90	`instrumentMSB` *	index, upper 16 bits	0	0	65535	65535	0
91	`sampleIDMSB` +	index, upper 16 bits	0	0	65535	65535	0

- "+" marks a generator that is only valid at the instrument level, and "*" one that is only valid at the preset level.
- "@" marks a generator that is read once, when the voice starts, and does not follow its modulators after that. The envelope generators may be read once as well (section 8.1.6).

The per-key directions of 82 and 83 are for rising keys: SoundFont 2 adds `value * (60 - key)` time-cents, so 1200 halves the time for each key upward.

"fs" stands for full scale, as in SoundFont 2: the excursion when the LFO or the envelope is at its full output. The "real units" columns give the same limits converted, so 1500 absolute cents is 20 Hz. Where a later clamp makes the range smaller, they give the limit that takes effect. The Q values of `filterQOffset` hold for the offset alone, with no `initialFilterQ` (section 7.1.1).

7.2 Modulator Source Enumerators

This section extends SoundFont 2 section 8.2.

An SFe synthesizer takes every source value as a whole number `x` from 0 to 16384, for all source types. The center of a source is exactly 8192 and its highest value exactly 16384, so a source reaches its full scale:

- A 7-bit value `v` is `128 * v` up to 64, and `8192 + 8192 * (v - 64) / 63` above 64, rounded to the nearest whole number. So 0 is 0, 64 is exactly 8192 and 127 is exactly 16384. This applies to the Note-On Velocity, the Note-On Key Number, the Poly Pressure, the Channel Pressure and every MIDI controller. A MIDI controller source is the 7-bit value of that controller alone: the controllers 33 to 63 are sources of their own and are not combined with the controllers 1 to 31 (section 11.3).
- A 14-bit value `w` is `w` up to 8192, and `8192 + 8192 * (w - 8192) / 8191` above 8192, rounded to the nearest whole number. So 8192 stays the center and 16383 is exactly 16384. This applies to the Pitch Wheel and to the two sources below.
- The Pitch Wheel Sensitivity is the 14-bit value `semitones * 128` rounded down, at most 16383.
- The Modulation Depth Range is the 14-bit value `min(16383, 8 * v)`, with `v` the 14-bit value of RPN 5 (section 11.4). So it is 8192 at 8 semitones, and the default range of 50 cents (`v` = 64) is 512, or 1/32.
- The No Controller source is 16384, the end of the range. With the positive direction it outputs exactly 1 for every built-in type and polarity, and a custom curve gives its value at 16384. The direction applies as for any other source, so with the negative direction it becomes 0, and a unipolar built-in type then outputs 0 and a bipolar one -1. A custom curve gives its value at 0. SoundFont 2 asks that No Controller never turn a modulator off, and SFe departs from that for the negative direction.

The Modulation Depth Range is a new general controller source with the index 17, next to the Pitch Wheel Sensitivity at 16. A synthesizer that plays SFe files must know it, or the modulation wheel has no vibrato.

The built-in source types of SoundFont 2 map `x` to their output as follows. With the negative direction, `x` first becomes `16384 - x` for every built-in type other than the Switch, as for a custom curve (section 7.2.1). A unipolar source uses `u = x / 16384`:

- **Linear** outputs `u`.
- **Concave** outputs `-(5/12) * log10(1 - u)`, limited to at most 1, and 1 when `u` is 1.
- **Convex** outputs `1 + (5/12) * log10(u)`, limited to at least 0, and 0 when `u` is 0.

A bipolar source uses `b = (x - 8192) / 8192`. Linear outputs `b`. Concave and convex output the unipolar curve mirrored around the center, `sign(b) * curve(|b|)`, so they are 0 at the center, -1 at the low end and 1 at the high end.

The Switch type outputs its maximum when `x` is 8192 or more and its minimum below that, so a 7-bit value of 64 is on. With the negative direction it is the exact opposite: the minimum when `x` is 8192 or more. The switch does not reverse `x`.

Apart from index 17, the general controllers that SoundFont 2 section 8.2.1 does not list stay reserved, and the MIDI controllers it calls illegal stay illegal (section 7.4). The general controller 127 (Link) is reserved, and a modulator that uses it must be ignored (section 7.4).

7.2.1 Custom Curves

In SFe, the source type field (bits 10 to 15 of the source enumerator) may hold the value 4, a custom curve. The index, the CC flag, the direction and the polarity select the source as in SoundFont 2. The curve itself is the one in the CURV sub-chunk given by the modulator's extended record (section 5.5.1, or section 5.3 for a default modulator).

The value of a source with a custom curve is computed as follows:

1. The source value is taken as the number `x` of section 7.2, from 0 to 16384.
2. If the direction is negative, `x` becomes `16384 - x`, as SoundFont 2 reverses a source (`1 - x / 16384`). So 64 stays exactly 8192, and 0 becomes 16384.
3. The curve is evaluated at `x`. Between two points, `y` is interpolated linearly. Below the first point, `y` is the first point's `y`, and above the last point it is the last point's `y`. Where several points share the same `x` (a vertical step), the last point with that `x` gives the value at that `x`.
4. A unipolar source outputs `y / 16384`, a bipolar source outputs `(y - 8192) / 8192`.

Unlike the SoundFont 2 curves, a bipolar custom curve is not mirrored around its center, so a curve can be asymmetric.

In a file that is not an SFe file, a modulator with source type 4 must be ignored, as SoundFont 2 asks for unknown source types. In an SFe file, a modulator must be ignored when its custom curve is invalid, or missing: a curve number of `dwCurveCount` or higher, or a file without `CURV` (section 5.4).

Two modulators with custom curves are identical (SoundFont 2 section 9.5.1) only if their curves have the same points.

7.3 Default Modulators

This section replaces SoundFont 2 section 8.4 for SFe files.

An SFe file must hold a `DMOD` sub-chunk in its INFO-list chunk, with the same records as `pmod` and a terminal record. When a default modulator has the extended flag, as one with a custom curve must, its extended record is in `XMOD` (section 5.3). The modulators of `DMOD` are the whole default list, so an SFe bank never depends on the defaults of the synthesizer that plays it. An SFe file without `DMOD` is not

valid, and a reader must reject it. A DMOD that holds only the terminal record or no bytes at all means that the bank has no default modulators. As in pmod, the last record of DMOD is the terminal record whatever it holds, and a reader drops it. The same goes for the last 16-byte record of XMOD and of the pmod and imod sub-chunks of xdta. A DMOD whose size is not a multiple of 10 bytes is malformed, and a reader must reject an SFe file with it (section 9.2 covers SoundFont 2 files).

The DMOD modulators act as the SoundFont 2 default modulators do. They apply at the instrument level, and an identical modulator in an instrument zone supersedes them. A modulator that is ignored supersedes nothing. When two modulators of DMOD are identical, the earlier one is ignored, as SoundFont 2 asks for two identical modulators in a zone.

The list below is the SFe default modulator list. Software that creates a new SFe bank must start it with this list in DMOD, and the author of the bank may then change it. It is also the list that a SoundFont 2 bank without a DMOD sub-chunk gets in the extended compatibility mode (section 9.2). Section 10 gives the reasons behind the list.

The transform is 0 (Linear) for all of them. The three modulators with a custom curve also have the extended flag, so their transform enumerator is 0x8000, and their records in XMOD give the indices of curves A and B in CURV. "Type" gives the source type, polarity and direction. The Pitch Wheel Sensitivity and the Modulation Depth Range are Positive Unipolar Linear sources.

Source	Source enum	Type	Amount source	Destination	Amount
Note-On Velocity	0x0502	Negative Unipolar Concave	-	initialAttenuation	960
CC 1 (modulation wheel)	0x0081	Positive Unipolar Linear	Modulation Depth Range (0x0011)	vibLfoToPitch	1600
CC 7 (volume)	0x0587	Negative Unipolar Concave	-	initialAttenuation	960
Pitch Wheel	0x020E	Positive Bipolar Linear	Pitch Wheel Sensitivity (0x0010)	fineTune	12700
CC 10 (pan)	0x028A	Positive Bipolar Linear	-	pan	500
CC 11 (expression)	0x058B	Negative Unipolar Concave	-	initialAttenuation	960
CC 91 (reverb)	0x00DB	Positive Unipolar Linear	-	reverbEffectsSend	1000
CC 93 (chorus)	0x00DD	Positive Unipolar Linear	-	chorusEffectsSend	1000
CC 73 (attack time)	0x12C9	Positive Bipolar Custom, curve A	-	attackVolEnv	9600
CC 75 (decay time)	0x12CB	Positive Bipolar Custom, curve B	-	decayVolEnv	9600
CC 72 (release time)	0x12C8	Positive Bipolar Custom, curve B	-	releaseVolEnv	9600
CC 74 (brightness)	0x02CA	Positive Bipolar Linear	-	initialFilterFc	4800
CC 71 (resonance)	0x02C7	Positive Bipolar Linear	CC 71, Positive Unipolar Switch (0x0CC7)	filterQOffset	250
CC 67 (soft pedal)	0x0CC3	Positive Unipolar Switch	-	initialFilterFc	-1200

Source	Source enum	Type	Amount source	Destination	Amount
CC 94 (variation)	0x00DE	Positive Unipolar Linear	-	`variationEffectsSend`	1000

The CC 71 modulator acts only above 64: its amount source is off below 64, and its source is 0 at 64 (section 10.3).

Section 10 gives the reasons for the main differences from the list of SoundFont 2.

The pitch wheel modulator keeps the amount 12700 of SoundFont 2, and with it a small error of SoundFont 2: a Pitch Wheel Sensitivity of `n` semitones gives the source `n * 128`, which is `n / 128` of full scale, so the wheel bends by 127/128 of `n` semitones both ways, up to a sensitivity of 64 semitones. A sensitivity of 2 bends 198.4 cents, and 12 bends 1190.6 cents.

The two custom curves are listed as `(CC value, y)` pairs. The `x` of each point is the source value of that CC value (section 7.2): `128 * value` up to 64, and above 64 the rounded `8192 + 8192 * (value - 64) / 63`, so 127 is 16384. Below 64 both shorten the time by 150 timecents per step. Above 64 they keep or lengthen it, and a higher value never makes the time shorter.

- Curve A (attack): (0, 0), (64, 8192), (69, 8192), (70, 8260), (72, 8453), (74, 8661), (77, 8988), (81, 9444), (87, 10149), (95, 11112), (105, 12337), (119, 14075), (127, 15074).
- Curve B (decay and release): (0, 0), (64, 8192), (65, 8263), (66, 8290), (67, 8297), (110, 8297), (111, 8309), (113, 8387), (114, 8434), (115, 8488), (116, 8548), (117, 8618), (118, 8697), (119, 8789), (120, 8896), (121, 9023), (122, 9176), (123, 9363), (124, 9601), (125, 9919), (126, 10384), (127, 11205).

7.4 Ignored Modulators

This section extends SoundFont 2 sections 8.2 and 8.3.

A reader must ignore a modulator that has any of these. The other modulators of the list stay. In SFe an ignored modulator supersedes nothing: it does not replace an identical modulator (SoundFont 2 section 9.5.1) of a default or global list.

- **A linked modulator.** SoundFont 2 lets a modulator use the output of another one: its source or its amount source is the general controller 127 (Link), or its destination has bit 15 set. SFe 4.0 reserves these. A later minor version may define links, so a reader of 4.0 must not half implement them.
- **An unknown transform.** The transform types are 0 (linear) and 2 (absolute value), as in SoundFont 2. In an SFe file the transform is the lower 15 bits of the transform enumerator, because bit 15 is the extended flag (section 5.5.1). Any other value is unknown, and a later minor version may define it (section 5.1).
- **An unknown destination.** The destination must be a generator of the version the reader knows (section 7.1). The SoundFont 2 numbers that are unused are unknown, and so are 60 (`endOper`) and the numbers above 91. In a file that is not an SFe file the generators above 60 are unknown too.
- **A destination that section 7.1 excludes:** `keyRange`, `velRange`, `keynum`, `velocity`, `instrument`, `sampleID`, `instrumentMSB` or `sampleIDMSB`, and, in a preset zone, a generator that is valid only at the instrument level.
- **An unknown source type** (in a file that is not an SFe file the custom curve type 4 is unknown too, section 7.2.1) and **a custom curve** that is invalid, or missing because its number is `dwCurveCount` or higher or the file has no `CURV` (sections 5.4 and 7.2.1).

- **An illegal or reserved controller** as the source or the amount source. The illegal MIDI controllers are those of SoundFont 2 section 8.2.1: 0, 6, 32, 38, 98 to 101 and 120 to 127. The controllers 33 to 63, except 38, are valid sources, although SoundFont 2 reserves them. A general controller (CC flag clear) is reserved unless its index is 0, 2, 3, 10, 13, 14, 16 or 17. In a file that is not an SFe file 17 is reserved too.
- **A problem with the extended record** (section 5.5.1).

When two modulators of one zone, or of `DMOD`, are identical as SoundFont 2 section 9.5.1 defines it (the transform included, and section 7.2.1 for custom curves), the earlier one must be ignored and the later one applies, as SoundFont 2 section 7.4 asks for a zone. The two are never added together.

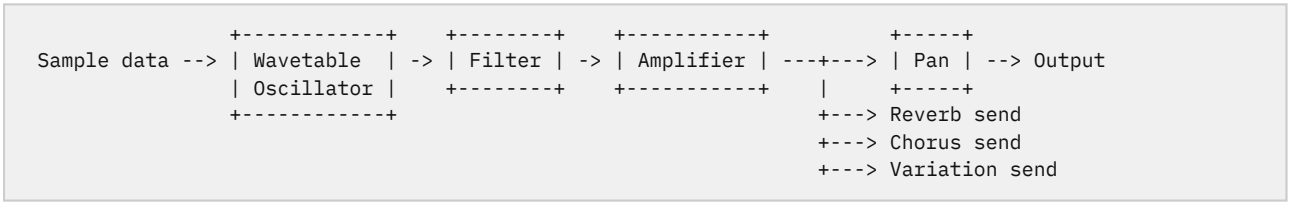
8. Parameters and Synthesis Model

This section extends SoundFont 2 section 9. Section 8.1 describes the SFe voice as a whole. Apart from the timing of "read once" in section 8.1, the sample points of section 8.1.1, the cutoff limit of section 8.1.2, the attenuation rule of section 8.1.3, the limits against artifacts of section 8.1 and the envelope rules and permissions of section 8.1.6 it adds no rules of its own, and the generator definitions in section 7.1.1 are the normative text.

8.1 Synthesis Model

This section extends SoundFont 2 section 9.1.

An SFe voice has the same signal path as a SoundFont 2 voice:



It is driven by two LFOs and three envelopes, where SoundFont 2 has two envelopes:

Source	Pitch	Filter cutoff	Volume (cB)	Amplitude (linear)
Modulation LFO	yes	yes	yes	SFe
Vibrato LFO	yes	SFe	SFe	SFe
Modulation Envelope	yes	yes	-	-
Second Modulation Envelope	SFe	SFe	-	-
Volume Envelope	-	-	yes	-

"SFe" marks a path that SFe adds. With every SFe generator at its default, all of these paths are off and the voice has the signal path of a SoundFont 2 voice. It still follows the rules of section 7.1 that differ from SoundFont 2, such as the unit of `initialAttenuation` and the zero time.

The whole voice (the oscillator, the three envelopes and both LFOs) starts `delayVoice` after its note on. Where this document says that a value is read once, when the voice starts, it is read at the note on, also when `delayVoice` delays the voice.

A synthesizer may add limits of its own where a value from the bank or a MIDI file would cause clicks or other artifacts. Examples are a minimum length of a note before a note off or an exclusive class ends it, a minimum attack or release time of the Volume Envelope, and the smoothing of changes to the

level, the pan or the filter. How long such a limit is, is left to the synthesizer. A delay, attack or hold time that the bank sets to zero time (-32768, section 7.1) stays zero time, so a voice whose delay and attack are zero time starts at its peak level without any smoothing.

8.1.1 Wavetable Oscillator

This section extends SoundFont 2 sections 9.1.1 and 9.1.2.

The oscillator is unchanged. In an SFe file `sampleModes` selects one of five ways to play the sample (section 7.1): no loop, a forward loop, a forward loop that ends at the note off, start on release, and the alternating loop.

The pitch is the SoundFont 2 pitch plus the excursion of the second Modulation Envelope (`modEnv2ToPitch`).

The sample points of a voice are its start, end, loop start and loop end after the address offset generators. A synthesizer must handle points that do not fit the sample:

- Each point is limited to the data of the sample itself, from its first sample data point to its end (the decoded data, for a compressed sample). The end and the loop end are exclusive, as in SoundFont 2, so they may equal the number of sample data points, and the start is at most the last sample data point.
- When the start is at or after the end, the voice plays no sample data in `sampleModes` 0 and 2, and in `sampleModes` 3 after the release. In a looping mode the rules below apply instead, so the loop still plays.
- When the loop end is before the loop start, the two are swapped.
- In a looping mode, a playback position at or past the loop end moves back by the length of the loop until it is inside the loop, so a start past the loop end plays the loop. For the alternating loop, positions count along its round trip of `2 * (end - start - 1)` sample data points (section 7.1) from the loop start, and the part after the loop end is the way back. So a start just past the loop end lands on the way back.
- While a looping mode loops, the end is not checked. The loop plays even when it lies partly before the start or after the end.
- After the loop of `sampleModes` 3 ends at the release, the voice plays until the end, and ends at once when its position is already at or past the end.
- A loop shorter than one sample data point is not played. A voice with the `sampleModes` value 1, 3 or 4 then plays as with 0.

8.1.2 Filter

This section extends SoundFont 2 section 9.1.3.

The filter stays a second order resonant filter. `filterType` selects its response, and the lowpass is the SoundFont 2 filter. The cutoff and the resonance are computed in this order:

1. `initialFilterFc` plus its modulators, clamped to 1500 to 13500 absolute cents.
2. The excursions of both LFOs (`modLfoToFilterFc`, `vibLfoToFilterFc`) and both Modulation Envelopes (`modEnvToFilterFc`, `modEnv2ToFilterFc`) are added.
3. The sum is limited to the ceiling `maxFilterFc`, unless the ceiling is 13500. It has no lower limit, and it should be limited to 0.45 times the sample rate (section 7.1.1, `filterType`).
4. `q` is `initialFilterQ` plus its modulators, clamped to 0 to 960 cB. `filterQOffset` plus its modulators is clamped to -960 to 960 cB by itself. The resonance `r` is the sum of the two, and it is not clamped

further.

5. The response of the selected type is computed from the cutoff and `r`, and scaled by `10^(-q / 400)`.

8.1.3 Amplifier

This section extends SoundFont 2 section 9.1.4.

The level of the voice is the product of these factors:

- The attenuation: `initialAttenuation` (plus its modulators, clamped to 0 to 1440 cB) minus `gain` (plus its modulators, clamped to -1440 to 1440 cB). The difference can be negative, so a voice can play above unity.
- The Volume Envelope.
- The LFO volume excursions in centibels (`modLfoToVolume`, `vibLfoToVolume`).
- The LFO amplitude factor (`modLfoToAmplitude`, `vibLfoToAmplitude`), which is linear and summed for both LFOs before one clamp to 0 to 2.

The attenuation must follow the modulators of `initialAttenuation` and `gain` for as long as the voice sounds, in every phase of the Volume Envelope, the sustain and the release included, apart from the smoothing that section 8.1 allows.

8.1.4 Effects Sends

This section extends SoundFont 2 section 9.1.5.

A voice has three sends: `reverbEffectsSend`, `chorusEffectsSend` and `variationEffectsSend`. Each is the degree to which the output of the voice is sent to its effects processor. The variation send is optional for the synthesizer.

8.1.5 Low Frequency Oscillators

This section extends SoundFont 2 section 9.1.6.

Both LFOs work the same way. After its delay (`delayModLFO`, `delayVibLFO`), an LFO starts at its start phase (`modLfoStartPhase`, `vibLfoStartPhase`) with its waveform (`modLfoWaveform`, `vibLfoWaveform`) and frequency. Its output then ramps linearly from zero to full over its fade time (`fadeModLFO`, `fadeVibLFO`), which scales all of its depths.

With the SFe generators at their defaults, an LFO is the SoundFont 2 triangle with no fade.

8.1.6 Envelope Generators

This section extends SoundFont 2 section 9.1.7.

The Volume Envelope has seven phases, one more than in SoundFont 2:

1. Delay (`delayVolEnv`).
2. Attack (`attackVolEnv`).
3. Hold (`holdVolEnv`).
4. Decay (`decayVolEnv`), down to the sustain level (`sustainVolEnv`).
5. Second decay (`decay2VolEnv`), a straight line in centibels from the sustain level to the end level (`sustainVolEnv + sustain2VolEnv`). With `sustain2VolEnv` at 0 this phase is skipped.
6. Sustain, at the end level, until the note off.
7. Release (`releaseVolEnv`), from the current level.

The second Modulation Envelope has the six phases and the generators of the first one, and runs independently of it.

Delay, attack and hold times of -32768 are zero time in all three envelopes (section 7.1). When the delay and the attack of the Volume Envelope are both zero time, the voice starts at its peak.

The attack of the Volume Envelope rises linearly in amplitude from silence to the peak, as SoundFont 2 describes it. The attack of both Modulation Envelopes follows the unipolar convex curve of section 7.2: with `t` the elapsed part of the attack, from 0 to 1, the envelope outputs `convex(t)`. The decay and the release of the Volume Envelope are straight lines in centibels, and those of the Modulation Envelopes are straight lines in their output, as in SoundFont 2.

A voice must end when its Volume Envelope reaches 960 cB of attenuation, in the release or in a sustain at that level. A synthesizer may end it earlier, once the envelope is so low that the voice is inaudible and its level can no longer rise: in the release, in the sustain, or in a second decay that does not rise. The rule looks at the envelope alone, so a voice with a large `gain` is still audible at that point and ends with a cut. A bank that raises `gain` should give its voices a release that reaches silence on its own.

With `ignoreNoteOff` at 1, a note off does not start the release, so the voice plays until its Volume Envelope or its sample ends.

A synthesizer may read the generators of the three envelopes once: the times and the levels when the voice starts, and the release time when the release starts. What a change of one of these generators does to an envelope that is already running is not defined, so a bank should not rely on a modulator that moves an envelope time or a sustain level during a note. This only concerns the envelope generators. The attenuation is not part of the Volume Envelope (section 8.1.3) and follows its modulators in every phase.

8.1.7 Modulation Interconnection Summary

This section extends SoundFont 2 section 9.1.8.

The three summing nodes of SoundFont 2 receive these paths, including the ones marked "SFe" in the table of section 8.1:

- **Pitch:** the Vibrato LFO, the Modulation LFO, the Modulation Envelope and the second Modulation Envelope.
- **Filter cutoff:** the Modulation LFO, the Vibrato LFO, the Modulation Envelope and the second Modulation Envelope. They are added after the clamp of `initialFilterFc` and before the ceiling (section 8.1.2).
- **Volume:** the Volume Envelope, the Modulation LFO and the Vibrato LFO, in centibels. The linear amplitude factor of the LFOs is applied on top (section 8.1.3).

9. Conversion Between Formats

In this section, "no amount source" means that the amount source enumerator is 0x0000, the No Controller source with the positive direction. Every rounding in this section is to the nearest whole number, and a half rounds toward positive infinity (2.5 becomes 3 and -2.5 becomes -2).

9.1 Conversion to SoundFont 2

When an SFe bank is written as SoundFont 2 or SoundFont 3, it goes through these steps:

- The form type must be `sfbk`, and the chunk sizes must be 32-bit (`RIFF`, not `RIFS`).
- `ifil` must be 2.04, or 3.0 when any sample is compressed.
- `wBank` must follow the SoundFont 2 conventions: 128 for a preset with the drum flag (a GM/GS drum preset), otherwise the bank MSB, or the bank LSB when the MSB is 0. SoundFont 2 holds one bank number, so this loses a part of the bank of a preset, as the list at the end of this section says.
- A bank in which a zone refers to an instrument or a sample above index 65535 cannot be written as SoundFont 2 (section 5.5), since SoundFont 2 has no `instrumentMSB` and `sampleIDMSB`.
- All generators above 60 must be dropped, along with every modulator (in zones and in `DMOD`) whose destination is one of them.
- A `sampleModes` generator with the value 4 must be written as 1, the forward loop, its closest SoundFont 2 form. Every other value outside 0 to 3 must be written as 0, the value SFe plays it as. SoundFont 2 reads the two lowest bits as flags, so a 5 would loop there. The generator is not dropped, so that the zone does not inherit the value of a global zone.
- The `isng` sub-chunk should not hold "SFe 4". "EMU8000" is the SoundFont 2 default.
- Every modulator that uses a custom curve must be dropped, and the `XMOD` and `CURV` sub-chunks must not be written.
- Every modulator that is ignored (section 7.4), in the zones and in `DMOD`, must be dropped, so that it does not become an active modulator of the SoundFont 2 file. An earlier one of two identical modulators may be kept, since SoundFont 2 ignores it too.
- The extended flag (section 5.5.1) must be clear in every modulator, so the transform enumerator holds the transform only.
- A name longer than 20 characters, or a bag, generator or modulator index or a `wSampleLink` above 65535, needs the `xdta-list` chunk (section 5.5), which SoundFont 2 files may hold too. A writer should write it then. One that does not must cut such names to 20 characters, and cannot write a bank whose indices do not fit 16 bits.
- The `pmod` and `imod` sub-chunks of `xdta`, if written, must hold only the terminal record, with the size of a modulator record (10 bytes).
- Every `initialAttenuation` generator value, at the preset level and at the instrument level, must be divided by 0.4 and rounded to the nearest whole number, so it is 2.5 times as large (SoundFont 2 stores the EMU8000 attenuation). The result must be limited to -32768 to 32767, the range of the SHORT. The amounts of the modulators that target it are not changed.
- A modulator that has CC 1 as its source, the Modulation Depth Range (0x0011, Positive Unipolar Linear) as its amount source and `modLfoToPitch` or `vibLfoToPitch` as its destination should be written without an amount source and with its amount divided by 32, rounded and limited to -1023 to 1023. CC 1 as its source means the controller CC 1 with any SoundFont 2 curve type, polarity and direction, and the modulator may have any transform. Every other modulator that uses the Modulation Depth Range, as its source or as its amount source (in any other form), must be dropped. A writer that does not convert the modulator above must drop it too.
- A reverb or chorus send modulator of the default form must be written with a fifth of its amount, rounded to the nearest whole number. The default form is CC 91 (0x00DB) with `reverbEffectsSend` or CC 93 (0x00DD) with `chorusEffectsSend`, Positive Unipolar Linear, no amount source and the Linear transform (0). No amount source means that the amount source enumerator is 0x0000 (the No Controller source). A crossed pair, such as CC 91 with `chorusEffectsSend`, is not of the default form. SoundFont 2 files store 200 where SFe stores a full send of 1000. This applies to `DMOD` and to every zone.

- The default modulators of the bank, after the steps above, should be written to a `DMOD` sub-chunk in the INFO-list of the `sfbk` file, with the record layout of `pmod` and a terminal record. The sub-chunk may be left out when the list of the bank, before the steps above, equals the list of section 7.3. Two lists are equal when they hold identical modulators (SoundFont 2 section 9.5.1, and section 7.2.1 for custom curves) in any order, each one the same number of times, with the same amounts. A file without `DMOD` gets the list of section 7.3 in the extended compatibility mode and the SoundFont 2 default modulators in the SoundFont 2 compatibility mode (section 9.2), so a writer that wants the same default modulators in both writes `DMOD` even then.

Some values are written as they are and play differently in a SoundFont 2 synthesizer, for example:

- A `fineTune` beyond -99 to 99 and an `exclusiveClass` above 127 are outside the SoundFont 2 ranges.
- A `sampleModes` value of 2 is played as no loop.
- A time of -32768 is played as 1 ms by a SoundFont 2 synthesizer that treats it as -12000.
- An attenuation above 1440 after the division is outside the SoundFont 2 range.
- The decay and release times of the Volume Envelope are timed against 960 cB, where SoundFont 2 names 100 dB, and a `sustainVolEnv` above 960 counts as 960 (section 7.1).
- The resonance peak of the lowpass filter is about 3 dB below the literal wording of SoundFont 2 (section 7.1.1).
- Two modulators of one list that differ in their transform (with any amounts) both apply in SFe. SoundFont 2 section 7.4 compares only the source, destination and amount source enumerators, so a SoundFont 2 synthesizer applies only the later one.
- A modulator source reaches its full scale at a 7-bit value of 127 (section 7.2), where SoundFont 2 stops at 127/128.
- A modulator from one of the controllers 33 to 63 is valid in SFe, and SoundFont 2 asks to ignore it.
- The No Controller source with the negative direction gives 0 (section 7.2), where SoundFont 2 keeps it at 1.

A SoundFont 2 file that is read and written again changes like this:

- Each attenuation value may change by one unit, once.
- A `sampleModes` value of 4 comes back as 1, and any other value outside 0 to 3 as 0.
- A send modulator of the default form with a file amount above 200 comes back as 200, and one with a file amount below -6554 comes back as -6554.
- A `wBank` above 128 comes back as section 9.1 writes it, from the bank that section 6.1 reads from it.
- Nothing else changes in what the file means in the extended compatibility mode (section 9.2).

The file is stable after the first writing.

An SFe bank that is written as SoundFont 2 and read again in the extended compatibility mode changes like this:

- The bank of a preset comes back as its `wBank` decoded by the layout of section 6.1. A preset with a bank MSB other than 0 loses its bank LSB. A preset with only a bank LSB comes back with that value as its bank MSB and with the bank LSB 0. A GM/GS drum preset comes back with its drum flag, the bank MSB 0 and the bank LSB 0. Presets that end up with the same program and `wBank` can no longer be told apart (section 11.2.1).
- The attenuation comes back exactly, from -13107 to 13107 (the file holds a 16-bit value).
- An alternating loop (`sampleModes` 4) comes back as a forward loop.

- The amount of a send modulator of the default form becomes a multiple of 5 and at most 1000, so 503 becomes 505. An amount of -32768 stays.
- The amount of a converted mod wheel vibrato becomes a multiple of 32, limited to -32736 to 32736.
- A mod wheel vibrato with no amount source and with an amount from -1023 to 1023 comes back with the Modulation Depth Range and 32 times the amount, so it follows RPN 5 afterwards.
- Everything that this section drops is gone, except when `DMOD` was left out, and except for what the steps of section 9.2 may add back: the envelope time and variation send modulators of section 7.3, and the CC 71 modulators with the amount 0 of the resonance step. A bank whose default modulators equal the list of section 7.3 may be written without `DMOD`, and reading the file as section 9.2 describes then gives the whole list back, including the modulators that SoundFont 2 cannot store.

9.2 Conversion from SoundFont 2

A synthesizer that plays SFe files loads a SoundFont 2 or SoundFont 3 file in one of two modes. It may offer both. The same goes for a file of an earlier draft that is read as a SoundFont 2 file (section 5.1).

- **SoundFont 2 compatibility mode.** The file is loaded as a SoundFont 2 file, the way the synthesizer loads such files without this document, and no step of this section is applied. Its default modulators are those of SoundFont 2 section 8.4. A synthesizer that knows the `DMOD` sub-chunk uses the modulators of that sub-chunk in their place, as they are written. This document defines nothing else about the mode: the unit of `initialAttenuation`, the amounts of the default modulators and the sample compression of SoundFont 3 stay as the synthesizer has them, and no other rule of this document applies to the file. That includes section 11: the MIDI behavior for a file in this mode is the one the synthesizer has for SoundFont 2 files. A SoundFont 2 file then plays as it always did in that synthesizer.
- **Extended compatibility mode.** The file is loaded by the steps of this section. Its default modulators are set as described under "Default modulators" below (the list of section 7.3 for a file without `DMOD`), and its effects sends are multiplied. The result is played as an SFe bank, so every rule of this document about how a bank plays applies to it, including section 11, and so does the `wBank` layout of section 6.1. The rules that section 7.1 gives for SFe files apply as well: the zero time rule, the `sampleModes` enumeration, the wider generator ranges and the 960 cB rules of the Volume Envelope. The default pitch wheel modulator of section 7.3 needs the wide range of `fineTune`, for example. A synthesizer that wants a SoundFont 2 file to play exactly as before uses the SoundFont 2 compatibility mode. The rules for files that are not SFe files (sections 5.5.1, 7.1, 7.2.1 and 7.4) apply to what the file stores. They do not apply to the modulators that the steps of this section add, supply or change, so a vibrato modulator that the modulation depth step gives the Modulation Depth Range is kept. This mode is recommended.

An SFe file is loaded exactly as this specification says in both modes. No step of this section ever applies to it, so its send amounts are never multiplied.

This specification does not define what a synthesizer does when it plays an SFe file and a SoundFont 2 file in the SoundFont 2 compatibility mode at the same time.

Software that converts a SoundFont 2 or SoundFont 3 file to an SFe file, or reads one into an SFe bank to edit it, must apply the required steps of this section and should apply the recommended ones. A synthesizer that loads a file in the extended compatibility mode must apply all of them, the recommended ones included, so that every synthesizer plays the file the same way in that mode. The reasons for the steps are given in section 10.

Software that converts a file to SFe must store the curves A and B of section 7.3 in `CURV` whenever its default modulators use them, and point the records in `XMOD` at them.

The steps are applied to the modulators and generators of the file (the zones and `DMOD`) in this order: initial attenuation, effects sends, modulation depth, the additions to the default modulators, and resonance. No step applies to the list of section 7.3.

Initial attenuation. Every `initialAttenuation` generator value, at the preset level and at the instrument level, must be multiplied by 0.4 and rounded to the nearest whole number, because an SFe file holds true centibels and a SoundFont 2 file holds the EMU8000 attenuation. The amounts of the modulators that target `initialAttenuation` are not changed. Each generator is rounded by itself, before the preset and the instrument value are added, so the sum can differ by a centibel from 0.4 times the sum of the file values.

Effects sends. The amount of a reverb or chorus send modulator of the default form (see section 9.1), in the default modulators and in every zone, must be multiplied by 5, and an amount above 1000 after that must become 1000. A result below -32768 must become -32768, so the amount stays within the SHORT range. A converted bank therefore has 1000 as its full send, like any SFe bank, and never has a send modulator of the default form above 1000.

Modulation depth. SoundFont 2 has no Modulation Depth Range source. A modulator with CC 1 as its source, no amount source, the destination `modLfoToPitch` or `vibLfoToPitch` and an amount from -1023 to 1023 should get the Modulation Depth Range (0x0011, Positive Unipolar Linear) as its amount source, and its amount should be multiplied by 32. CC 1 as its source means the controller CC 1 with any SoundFont 2 curve type, polarity and direction, and the modulator may have any transform. This applies to the default modulators and to every zone. Any other modulator is left alone and does not follow RPN 5. A CC 1 vibrato modulator in a zone with an amount beyond -1023 to 1023 is such a modulator, so it is no longer identical to the default vibrato modulator after the conversion, and it adds to it where it replaced it in the SoundFont 2 file.

Default modulators. What the default modulators of a converted bank are depends on the file:

- A `DMOD` sub-chunk whose size is not a multiple of 10 bytes is treated as missing, so the file counts as a file without a `DMOD` sub-chunk in every step of this section.
- A file with a `DMOD` sub-chunk keeps its list, converted by the steps above, and gets the additions below. A `DMOD` that holds only the terminal record or no bytes at all is empty: it means no default modulators, and nothing is added to it. A `DMOD` that holds modulators that are all ignored (section 7.4) is not empty, and gets the additions.
- A file without a `DMOD` sub-chunk gets the list of section 7.3, exactly as it is. No step of this section and no addition is applied to that list.

The additions are these. A modulator that is ignored (section 7.4) does not count for the tests of the additions below.

- The three envelope time modulators (CC 73, CC 75 and CC 72) of section 7.3 should be added, unless the list has a modulator with CC 72, CC 73 or CC 75 as its source or amount source.
- The CC 94 to `variationEffectsSend` modulator should be added, unless the list has a modulator with CC 94 as its source or amount source.

Resonance. The modulators of the file that target `initialFilterQ` are left as they are, and no modulator to `gain` is added for them. In a file without a `DMOD` sub-chunk, every instrument zone, the global zone included, that has a modulator with CC 71 as its source (with any SoundFont 2 curve type, polarity and direction, any amount source and any transform) and the destination `initialFilterQ` should

get one more modulator: CC 71 Positive Bipolar Linear (0x02C7), the amount source CC 71 Positive Unipolar Switch (0x0CC7), the destination `filterQOffset`, the Linear transform (0) and the amount 0. It is identical, as SoundFont 2 section 9.5.1 defines it, to the CC 71 modulator of section 7.3 and replaces it, so such a zone gets only the resonance of the file. A modulator that is ignored (section 7.4) does not count. A preset zone gets nothing, so a CC 71 modulator of a preset zone adds to the one of section 7.3 (section 10.3).

Conversion to and from formats other than SoundFont 2 or SoundFont 3 is not defined at this time.

10. Design Rationale

This section is informative. It explains why the default modulators of section 7.3 and the conversion steps of section 9 are the way they are.

10.1 Default Modulators in Every File

A SoundFont 2 bank relies on the default modulators of the synthesizer, and synthesizers do not agree on them. An SFe file holds its whole list in `DMOD`, so it sounds the same everywhere, and a bank can change or remove any default modulator without the synthesizer adding it back.

10.2 Effects Send Amounts

The SoundFont 2 defaults send 20 % to the reverb and the chorus at the highest controller value. In practice synthesizers play a much higher send, and existing banks are balanced by ear against that. SFe therefore stores the real amount, with a default of 1000 (a full send at the controller value 127).

Section 9.2 multiplies the SoundFont 2 amounts by 5 and stops at 1000 in the extended compatibility mode. The amounts of a file were balanced against a default of 200, so they are raised together with the default. The reasons for stopping at 1000 are these:

- A bank that lowers the send slightly from 200, for example to 150, would be almost dry next to a default of 1000. Converted, it becomes 750, which is still lower than the default and still audible.
- A bank that already assumes a full send of 1000 is never played louder than that. An amount of 800 in such a bank becomes 1000, the same as the default, and no more.
- An amount of 0 stays 0, so a bank can still turn the sends off.

Stopping at 1000 cannot be undone, so a converted bank that is written as SoundFont 2 again has 200 where the file had more. That is accepted: SoundFont 2 is a conversion target for an SFe bank, and the written file sounds the way the bank does.

10.3 Resonance

The SoundFont 2 filter lowers its level by half of `initialFilterQ` (`q / 20` dB), so that a resonant peak does not clip. A controller that changes the resonance during a song should not change the level of the part. So the default CC 71 modulator targets `filterQOffset`, which moves the resonance without the level, and needs no modulator to `gain` next to it.

It only acts above 64. Below 64 a resonance controller of a GS or XG device takes away the resonance of a patch, down to about a flat response, and leaves a patch without resonance as it is. A single modulator cannot follow the resonance of each patch that way: a bipolar one would push a patch without reson-

ance below the Butterworth response and audibly darken it, and many MIDI files that target these devices set the resonance a little below 64 at their start. So the default leaves the values below 64 alone, and a bank that wants them to lower its resonance brings a CC 71 modulator of its own.

A converted SoundFont 2 file keeps its own modulators. A CC 71 to `initialFilterQ` modulator of a zone lowers the level with the resonance, as it does in a SoundFont 2 synthesizer. It targets another generator than the default CC 71 modulator, so it would add to it, and the zone would get about twice the resonance range that the file asks for. The resonance step of section 9.2 turns the default off with an amount of 0 in every instrument zone that has its own modulator, and a global zone covers its local zones. A modulator only in a preset zone still adds to the default, since a preset zone can only add to the instrument. The `DMOD` of a SoundFont 2 file gets no CC 71 modulator, so the file plays as it does in other synthesizers that read `DMOD`.

10.4 Envelope Time Controllers

CC 73, CC 75 and CC 72 make the attack, decay and release shorter below 64 and longer above it, by different amounts on each side. A SoundFont 2 curve cannot describe that, and it took several modulators and special handling in the synthesizer. A custom curve on a bipolar source does it with one modulator for each controller, which a bank can read, change or replace like any other.

The two curves are a middle ground between the responses of General MIDI hardware. A bank that needs another response brings its own curves in `CURV` and its own modulators in `DMOD`.

SoundFont 2 cannot store these modulators, so section 9.2 adds them to the `DMOD` list of a converted file. A file without `DMOD` gets the list of section 7.3, which holds them already. A file that has a CC 72, CC 73 or CC 75 modulator of its own has made its choice and keeps it. The same goes for CC 94 and the variation send.

10.5 Pan Amount

The pan source is bipolar, so an amount of 500 reaches the ends of the `pan` range: -500 at a CC 10 value of 0 and 500 at 127, with 64 at the center (section 7.2). The 1000 of SoundFont 2 section 8.4.6 would reach them at a quarter and three quarters of the controller's travel. A real SoundBlaster card plays the 500 response.

10.6 Velocity to Filter Cutoff

The Note-On Velocity to `initialFilterFc` default modulator of SoundFont 2 is left out. Most SoundFont 2 synthesizers do not apply it, and banks are made with that in mind. With it, quiet notes sound muffled.

10.7 Brightness

CC 74 moves the cutoff by 4800 cents to each side, so four octaves down at 0 and four octaves up at 127. That is 75 cents per step below 64 and about 76 above it, close to General MIDI hardware: a Yamaha XG engine moves the cutoff by about 75 cents per step, and a Roland GS engine by about 100 cents per step, less near the top of its range. Twice that range would close the filter far too early in a file that sweeps CC 74.

10.8 Modulation Depth Range

GM2 defines RPN 5 as the depth of the modulation wheel vibrato. A synthesizer that scales the vibrato by itself has to guess which modulators of a bank are the vibrato. As a source, the range is part of the bank: the default modulator follows RPN 5, and a bank can use the range for something else or ignore it.

The source is full at 16 semitones and not at the 128 that RPN 5 can express. So the amount that stands for 50 cents is 1600, a converted SoundFont 2 amount is multiplied by 32, and amounts up to 1023 fit a SHORT. GM2 recommends a range of at most 6 semitones.

10.9 Channel Pressure

The Channel Pressure to `vibLfoToPitch` default modulator of SoundFont 2 is left out. The GM, GS and XG sound modules do not answer the channel pressure with vibrato unless a file sets that up, so music made for them plays with a vibrato it never had. A bank that wants the vibrato adds the modulator to its `DMOD`.

11. MIDI Behavior

A bank author has to know what a MIDI file does to a bank, so this section defines the part of the MIDI behavior that decides it. A synthesizer must follow all of it, except where a rule says "should" or "may", and except for a SoundFont 2 file in the SoundFont 2 compatibility mode (section 9.2). Everything else is up to the synthesizer: portamento, mono mode, the channel and master tuning and the tuning messages, velocity sensitivity, the effects and their parameters, the GS and XG controller routing, random panning and the master parameters (section 11.6.1).

A synthesizer may support MIDI features that this specification does not define, such as these. Such a feature may be switched on only by messages, values or settings that this specification does not define, or by the part of a defined message that it leaves undefined (the mono mode of CC 126, for example). A defined message must still do everything this specification requires of it. A MIDI file that switches on none of these features must play exactly as this specification requires, with the synthesizer at its default settings.

The effect sends of a voice on a drum channel are not defined by SFe 4.0. GS and XG devices set them for each key of a drum kit, so a synthesizer may change the reverb, chorus and variation sends of such a voice per key, also when no message asks for it. This is an exception to the extension clause above, and a bank cannot rely on its send generators and modulators on a drum channel. Every other parameter of a single drum key (its pitch, level or pan, for example) follows the extension clause.

> **Note**
>
> The parameters of single drum keys are planned for SFe 4.1.

A synthesizer must receive every message of this section with any device ID and in every mode, except that in GM mode it ignores the bank selects, the non-registered parameters and every GS and XG System Exclusive message of this section apart from the mode messages of section 11.1, as General MIDI devices do. "Channel 10" means the tenth channel of every group of 16 channels. In the byte lists,

`dd` is the device ID, `n` a device number and `cs` the Roland checksum: 128 minus the sum of the address and data bytes modulo 128, and 0 when that sum is a multiple of 128. A synthesizer may ignore the checksum of every Roland message in this section.

11.1 Modes

A synthesizer is in one of four modes: GM, GS, GM2 or XG. It starts in GS. The mode decides how presets are selected (section 11.2), what the part mode messages do (section 11.6) and what GM ignores. It also sets the default part mode value of channel 10 (section 11.6) and, in XG, the drum part protection of channel 10 (section 11.2.2). These messages set it, and each of them is a system reset:

Message	Bytes	Mode
GM1 System On	`F0 7E dd 09 01 F7`	GM
GM System Off	`F0 7E dd 09 02 F7`	GS
GM2 System On	`F0 7E dd 09 03 F7`	GM2
GS Reset	`F0 41 dd 42 12 40 00 7F 00 41 F7`	GS
GS Exit	`F0 41 dd 42 12 40 00 7F 7F 42 F7`	GM
GS System Mode Set	`F0 41 dd 42 12 00 00 7F 0m cs F7`	GS
XG System On	`F0 43 1n 4C 00 00 7E 00 F7`	XG
XG All Parameter Reset	`F0 43 1n 4C 00 00 7F 00 F7`	XG

In the GS System Mode Set message, `m` is 0 for MODE-1 or 1 for MODE-2. Both values set the mode GS and reset the synthesizer as described below. Any other value is ignored.

A system reset stops every voice (a synthesizer may end them at once or let them release), puts every channel into the state of sections 11.2.2, 11.3 and 11.4, and sets the part mode value of section 11.6 back to its default.

11.2 Preset Selection

This section has no counterpart in SoundFont 2. It defines which preset a MIDI channel plays.

A synthesizer must follow sections 11.2.1 and 11.2.2 and rules 1 and 2 of section 11.2.3. The fallbacks of rule 3 are a recommendation.

11.2.1 Kinds of Preset

A preset is identified by four values: its program (`wPreset`, 0 to 127), its bank MSB, its bank LSB and its drum flag (section 6.1).

- A **GM/GS drum preset** has the drum flag set.
- An **XG drum preset** has the drum flag clear and a bank MSB of 120 or 127, in a file that passes the XG test.
- A **drum preset** is either of the two.
- A **melodic preset** is every other preset. That includes a preset at bank MSB 126 (the XG SFX kits) and a preset at bank MSB 120 or 127 in a file that fails the XG test.

A file passes the **XG test** when it has at least one preset at bank MSB 120 or 127, and every such preset (with or without the drum flag) has one of these programs: 0 to 9, 16, 17, 24 to 33, 40, 41, 48, 56, 57, 58, 64, 65, 66, 126 or 127. These are the drum kit numbers that XG devices use, counted from 0. When several files are loaded together, each file is tested by itself.

> **Note**
>
> The test exists because many SoundFont 2 banks hold melodic Roland MT presets at bank 127.

The presets are ordered by program, then bank MSB, then bank LSB, with all GM/GS drum presets after the others. "The first" in this section means the first in this order. When two presets of a file have the same four values, the one that comes first in `PHDR` is used and the other cannot be selected.

When several files are loaded together, the synthesizer gives them an order of priority. Of the presets with the same four values, only the one from the file with the highest priority can be selected. The presets that remain from all files form one list, ordered as above, and every rule of this section works on that list.

11.2.2 Channel State and Modes

A channel holds a program, a bank MSB, a bank LSB and a drum flag. CC 0 and CC 32 only store their value. A Program Change stores the program and selects a preset from the four values and the mode of the synthesizer. A system reset selects again, and a part mode message may select (section 11.6).

A channel is a drum channel exactly when its selected preset is a drum preset, whatever its drum flag says. A channel without a selected preset plays nothing, so this specification does not say whether it is a drum channel. What else a drum channel changes in a synthesizer is not defined here.

The mode of the synthesizer (section 11.1) decides how the four values are used:

Mode	Default bank MSB	The variation is chosen by	A drum channel is chosen by
GM	0	nothing, the bank stays 0	the drum flag, on channel 10
GS	0	the bank MSB, the LSB is a map	the drum flag, on channel 10 or by SysEx
GM2	121	the bank LSB	the bank MSB 120 or 127
XG	0	the bank LSB	the bank MSB 127 or 120

- In GM, CC 0 and CC 32 are ignored, so the bank MSB and LSB of a request are always 0. Apart from that GM follows the rules of GS.
- The drum bank MSB is 127 in XG and 120 in GM2.
- On a system reset every channel gets program 0, bank LSB 0 and the default bank MSB of the mode. Channel 10 of every group of 16 channels is the drum channel: in GM and GS its drum flag is set, and in XG and GM2 its drum flag is clear and its bank MSB is the drum bank MSB of the mode. Every other channel has the drum flag clear. In XG and GM2 rule 1 of section 11.2.3 only guards against a drum flag that is set some other way.
- On XG channel 10 a melodic part mode value must be ignored and not stored (section 11.6), and so must a CC 0 message with a bank MSB other than 120, 126 and 127. This is the drum part protection of XG devices, which keeps channel 10 on the drum banks. A request with the bank MSB 126 and an exact match plays that preset, which is melodic, so the channel is then not a drum channel. A request with the bank MSB 126 and no exact match belongs to the list of rule 3.2 of section 11.2.3 and not to the one of rule 3.3, so it should get a drum preset whenever the file has one. A synthesizer that selects a fallback here should not select a melodic preset while the file has a drum preset. In GM2 channel 10 can become melodic: a bank MSB other than 120 and 127, such as 121, makes it melodic like any other channel.
- A synthesizer may let the user turn the drum part protection of XG channel 10 off, as XG devices do. It is on by default. When it is off, channel 10 takes the messages above like any other channel,

and the rules above about the bank MSB 126 on XG channel 10 do not apply to it.

- In GS (GM ignores the bank selects), on a channel with the drum flag clear, a bank MSB of 120 or 127 selects an XG drum preset only by an exact match (rule 2), and only in a file that passes the XG test. A channel with the drum flag set may fall back to an XG drum preset (rule 3.1). In XG and GM2 the drum flag is not used.

11.2.3 Selecting a Preset

The request is the program, the bank MSB, the bank LSB and the drum flag of the channel. List 3.2 below also depends on the channel itself (XG channel 10) and on its drum part protection.

1. In XG and GM2, a channel with the drum flag set has the flag cleared before the preset is selected. Its bank LSB becomes 0 and its bank MSB becomes the drum bank MSB of the mode.
2. A preset with the same four values as the request is selected. In XG and GM2, when the requested bank MSB is 120 or 127, this preset must be an XG drum preset. Otherwise it is not selected, and the search continues at rule 3.
3. Without such a preset, a synthesizer should use one of the three lists below, chosen by the request. In it, the first rule that finds a preset selects it. A file without presets selects nothing.
    1. **The drum flag is set** (GM and GS):
        1. A GM/GS drum preset with the same program and bank LSB.
        2. A GM/GS drum preset with the same program.
        3. A drum preset with the same program, if the program is below 49 or the preset has the bank MSB 120.
        4. The first GM/GS drum preset.
        5. The first drum preset.
        6. The first preset.
    2. **XG or GM2 with a bank MSB of 120 or 127, or XG channel 10 with a bank MSB of 126 while the drum part protection is on:**
        1. An XG drum preset with the same program.
        2. A drum preset with the same program, if the program is below 49 or the mode is GM2.
        3. The first XG drum preset.
        4. The first drum preset.
        5. The first preset.
    3. **Every other request.** The candidates are the melodic presets with the same program. Without a candidate the first preset is selected. Otherwise:
        1. GM and GS: the candidate with bank MSB 0 and the same bank LSB.
        2. GM and GS: the candidate with the same bank MSB. XG and GM2: the candidate with the same bank LSB.
        3. The candidate whose bank MSB or bank LSB equals the higher of the requested MSB and LSB. This rule is skipped in XG and GM2 when the requested bank LSB is 64.
        4. The first candidate.

Where a rule matches several presets, the first of them is selected.

Without a preset that rule 2 selects, a synthesizer may select another preset than the lists give, or none. A channel without a selected preset plays nothing until a later selection finds one. Either way the program, the bank MSB and the bank LSB of the channel stay as they were requested, apart from the changes of rule 1.

Note

- Bank MSB 126 is not a drum bank, because the XG SFX kits are rarely what a file wants. An exact match still selects them, as a melodic preset. Only on XG channel 10 a request with no exact match uses the list of rule 3.2, so it should select a drum preset whenever the file has one, because a part mode value cannot make that channel melodic while the drum part protection is on.
- Above program 48 the kit numbers of GS and XG no longer mean the same. Program 56 is the SFX set in GS and an ordinary kit on later XG devices. GM2 numbers its kits as GS does, so rule 3.2.2 has no such limit in GM2, and rule 3.1.3 takes a kit at bank MSB 120, the GM2 drum bank, at any program.
- In GS the bank LSB selects the map of a device, so a missing variation should fall back to the capital tone of its own map, the preset at bank MSB 0 with the same program (rule 3.3.1).
- Bank MSB 64 holds the XG SFX voices. Without the exception, rule 3.3.3 would send an XG request for the variation at bank LSB 64 to an SFX voice at bank MSB 64, so the rule is skipped for a bank LSB of 64.

11.3 Controllers

Every MIDI controller, apart from the illegal controllers of section 7.4, is a 7-bit modulator source. The controllers 33 to 63, which MIDI 1.0 names as the LSBs of the controllers 1 to 31, are not combined with them: each of them except 38 is a source of its own, and a message for one of the two never changes the value of the other. MIDI files almost never send these LSBs for modulation, and a modulator that wants a finer control can use one of them as its own source. The data entry (CC 6 with CC 38, section 11.4) and the bank select (CC 0 with CC 32, section 11.2.2) do use their LSB. None of these four is a modulator source, as SoundFont 2 section 8.2.1 lists them among the illegal controllers (section 7.4).

A bank is made for a known starting state: a bipolar modulator from CC 74 leaves the cutoff alone only when CC 74 starts at 64. So a synthesizer must give the modulator sources these values when it starts and on a system reset:

Controller	Name	Default
CC 7	Volume	100
CC 8	Balance	64
CC 10	Pan	64
CC 11	Expression	127
CC 71	Resonance	64
CC 72	Release Time	64
CC 73	Attack Time	64
CC 74	Brightness	64
CC 75	Decay Time	64
CC 76	Vibrato Rate	64
CC 77	Vibrato Depth	64
CC 78	Vibrato Delay	64
CC 81	General Purpose Controller 6	64

Controller	Name	Default
CC 83	General Purpose Controller 8	64
CC 91	Reverb Send	40

Every other controller that can be a modulator source defaults to 0, the controllers 33 to 63 included.

The other sources:

Source	Default
Pitch Wheel	8192, the center
Pitch Wheel Sensitivity	2 semitones
Modulation Depth Range	50 cents
Channel Pressure	0
Poly Pressure	0, for every key

> **Note**
>
> The reverb send of 40 is the default of GM2, GS and XG. With the default modulators of section 7.3 it is a send of about a third, so a file that never sends CC 91 still has some reverb, as on the devices it was made for.

11.4 Registered Parameters

A registered parameter is selected with CC 101 (MSB) and CC 100 (LSB) and set with the data entry, CC 6 (MSB) and CC 38 (LSB). The data value is `v = msb * 128 + lsb`. A data entry is applied when either of the two arrives, with the other one as it is stored.

RPN (MSB, LSB)	Name	Value	Default
0x00, 0x00	Pitch Wheel Sensitivity	the data MSB in semitones plus the data LSB in cents (1)	2 semitones
0x00, 0x05	Modulation Depth Range	the data MSB in semitones plus the data LSB in 100/128 cents	50 cents
0x7F, 0x7F	Null	no parameter is selected, and a data entry does nothing	selected

- (1) The Pitch Wheel Sensitivity has no maximum apart from the limit of its source (section 7.2). A data LSB above 99 still adds that many cents.
- The Null parameter is 0x7F, 0x7F. Every parameter number that the table does not list is not defined by this specification (section 11), the channel fine and coarse tuning (0x00, 0x01 and 0x00, 0x02) among them.
- The Pitch Wheel Sensitivity and the Modulation Depth Range are modulator sources (section 7.2). They have no effect other than through the modulators of the bank.
- Selecting a registered or non-registered parameter sets the stored data entry, MSB and LSB, to 0. The parameter that was selected last, registered or non-registered, is the one a data entry sets. In GM a non-registered parameter is still selected this way, and a data entry for it then does nothing.
- The data increment and decrement (CC 96 and CC 97) are not defined here.
- A synthesizer may implement the non-registered parameters of SoundFont 2 section 9.6. When it does, they address every generator of section 7.1 by its number, the SFe generators included.

The following non-registered parameters, selected with CC 99 (MSB) and CC 98 (LSB), set a controller. The data MSB becomes the value of the controller:

NRPN (MSB, LSB)	Controller
0x01, 0x08	CC 76, vibrato rate
0x01, 0x09	CC 77, vibrato depth
0x01, 0x0A	CC 78, vibrato delay
0x01, 0x20	CC 74, brightness
0x01, 0x21	CC 71, resonance
0x01, 0x63	CC 73, attack time
0x01, 0x64	CC 75, decay time
0x01, 0x66	CC 72, release time

11.5 Channel Mode and Pedal Controllers

A Note On with the velocity 0 is a note off, as MIDI 1.0 says.

CC 64 is also a modulator source (section 11.3), and a modulator from it follows each change, as for any other controller. CC 120 to 127 are not modulator sources (section 7.4).

Controller	Name	What it does
CC 64	Sustain Pedal	At 64 or above, a note off holds its voices. Below 64 they are released.
CC 120	All Sound Off	Ends every voice of the channel at once, without its release (section 8.1 allows smoothing the cut).
CC 121	Reset All Controllers	Resets the controllers as RP-015 says, see below.
CC 123	All Notes Off	A note off for every voice of the channel, see below.
CC 124 and 125	Omni Off and On	As All Notes Off.
CC 126 and 127	Mono and Poly On	As All Notes Off, as MIDI 1.0 asks. Mono mode is not defined here.

- All Notes Off acts as a note off for every voice of the channel. So while the sustain pedal is down the voices are held, and they are released when it goes up.
- All Notes Off also releases the voices with `ignoreNoteOff` at 1, which a note off would not release, even while the sustain pedal is down.
- A voice with `ignoreNoteOff` at 1 is not held by the sustain pedal.
- Reset All Controllers follows the MIDI recommended practice RP-015. It puts the pitch wheel to its center, the channel and poly pressure to 0, CC 1 and CC 33 to 0, CC 11 to 127 and CC 43 to 0, CC 64 to 67 to 0 and the selected parameter numbers to null. It does not change the bank selects, the program, CC 7, CC 10, CC 70 to 79, CC 91 to 95 or the values set through registered parameters. Every other controller keeps its value too, as GS and XG devices do and as RP-015 allows when it is documented. When it lowers CC 64 below 64, the voices that the sustain pedal holds are released, as with a CC 64 message.
- CC 0, CC 32, CC 6, CC 38 and CC 98 to 101 are described in sections 11.2 and 11.4. Portamento (CC 5, CC 65 and CC 84), the sostenuto pedal (CC 66) and the data increment and decrement (CC 96 and CC 97) are not defined here and are left to the synthesizer. As far as this specification defines, every other controller acts only through the modulators of the bank.

11.6 System Exclusive Aliases

These messages do what a channel message does. A synthesizer must treat each of them exactly as the message or parameter it stands for, so a value sent either way gives the same sound. The value 0 of the two part pan messages is not defined (section 11.6.4).

An alias of a registered parameter sets the value of that parameter directly, and BEND PITCH CONTROL sets it with 0 cents. It does not change which parameter is selected, and it leaves the stored data entry alone, so a later data entry still goes to the parameter that CC 101 and CC 100 selected.

A GS message is `F0 41 dd 42 12`, three address bytes, the data, `cs F7`. An XG message is `F0 43 1n 4C`, three address bytes, the data, `F7`. This section only defines a message that starts at a listed address and carries the data of that one parameter. A message that starts inside a parameter, or that carries data for several parameters, is not defined here.

The part mode is the same for both sets of part parameters. A channel keeps one part mode value: 0 means melodic and any other value means drum. GS USE FOR RHYTHM PART (section 11.6.2) and XG PART MODE (section 11.6.3) both set it, in GS, GM2 and XG (GM ignores both messages), so the last of the two decides. After a system reset it is 0, except on channel 10, where it is 2 in XG and 1 in the other modes. What a message does after storing its value depends on the mode:

- In GS, when the value differs from the value stored before the message, or the drum flag that the new value implies (set for a drum value, clear for a melodic one) differs from the drum flag of the channel, the drum flag is set or cleared that way, the bank MSB and LSB become 0, and the channel stores program 0 and selects it. A message that changes neither does not select again.
- In XG and GM2, the drum flag is cleared. A drum value sets the bank MSB to the drum bank MSB of the mode unless it already is 120 or 127, and a melodic value sets it to the default bank MSB of the mode. The bank LSB stays. Then:
  - A drum value on a channel that was not a drum channel before the message stores program 0 and selects it.
  - Otherwise, unless the value is a drum one and the bank MSB already was 120 or 127, the stored program of the channel is selected again.
  - Whether a channel without a selected preset (section 11.2.3) was a drum channel before the message is not defined.
- On XG channel 10 a melodic value must be ignored and not stored while the drum part protection is on (section 11.2.2).

11.6.1 Master Parameters

This specification does not define master parameters, such as a master volume, tuning, key shift, pan or balance, or the messages that set them (the universal Master Volume, Master Balance, Master Fine Tuning and Master Coarse Tuning, and the GS and XG master messages). They do not depend on the sound bank, and are left to the synthesizer.

11.6.2 GS Part Parameters

The address is `40 1x aa`, `40 2x aa` or `40 4x aa`. `x` selects the part: 0 is channel 10, 1 to 9 are channels 1 to 9 and A to F are channels 11 to 16. An address that starts with `50` in place of `40` selects the same part of the second group of 16 channels.

Address	Name	Stands for
`1x 00`	TONE NUMBER	CC 0 with the first data byte, then a Program Change with the second
`1x 15`	USE FOR RHYTHM PART	the part mode value (section 11.6), 0 is melodic and any other value is a drum channel. In XG and GM2 it acts as the XG PART MODE of section 11.6.3
`1x 19`	PART LEVEL	CC 7
`1x 1C`	PART PANPOT	CC 10
`1x 21`	CHORUS SEND LEVEL	CC 93
`1x 22`	REVERB SEND LEVEL	CC 91
`1x 2C`	DELAY SEND LEVEL	CC 94
`1x 30`	TONE MODIFY 1	CC 76, vibrato rate
`1x 31`	TONE MODIFY 2	CC 77, vibrato depth
`1x 32`	TONE MODIFY 3	CC 74, brightness
`1x 33`	TONE MODIFY 4	CC 71, resonance
`1x 34`	TONE MODIFY 5	CC 73, attack time
`1x 35`	TONE MODIFY 6	CC 75, decay time
`1x 36`	TONE MODIFY 7	CC 72, release time
`1x 37`	TONE MODIFY 8	CC 78, vibrato delay
`2x 04`	MOD LFO1 PITCH DEPTH	RPN 5 with `v = round(6.4 * value)`, which is `5 * value` cents rounded to a step of RPN 5 (1)
`2x 10`	BEND PITCH CONTROL	RPN 0, `value - 64` semitones, at least 0 and at most 24
`4x 00`	TONE MAP NUMBER	CC 32

(1) Roland devices scale this parameter as 0 to 600 cents over 0 to 127, about 4.72 cents per step. SFe uses 5 cents per step, so that the default value of 10 gives exactly the 50 cents of the RPN 5 default.

11.6.3 XG Part Parameters

The address is `08 pp aa`, with `pp` the part from 0 to 63. Part `pp` is channel `pp + 1`.

Address aa	Name	Stands for
`01`	BANK SELECT MSB	CC 0
`02`	BANK SELECT LSB	CC 32
`03`	PROGRAM NUMBER	Program Change
`07`	PART MODE	the part mode value (section 11.6), 0 is melodic and any other value is a drum channel
`0B`	VOLUME	CC 7
`0E`	PAN	CC 10
`12`	CHORUS SEND	CC 93
`13`	REVERB SEND	CC 91
`14`	VARIATION SEND	CC 94

Address aa	Name	Stands for
15	VIBRATO RATE	CC 76
16	VIBRATO DEPTH	CC 77
17	VIBRATO DELAY	CC 78
18	FILTER CUTOFF	CC 74
19	FILTER RESONANCE	CC 71
1A	EG ATTACK TIME	CC 73
1B	EG DECAY TIME	CC 75
1C	EG RELEASE TIME	CC 72
20	MW LFO PMOD DEPTH	RPN 5 with `v = round(6.4 * min(value, 40))`, which is `5 * value` cents rounded to a step of RPN 5, at most 200 cents
23	BEND PITCH CONTROL	RPN 0, `value - 64` semitones, at least 0 and at most 24

11.6.4 Random Pan

The value 0 of GS PART PANPOT and of XG PAN is not defined by this specification. Some devices play it as a random pan position for every note, a feature that section 11 allows.