The format of Cakewalk Instrument definition files
and
A guide to creating instrument definitions

Heiko Plate: http://www.heikoplate.de/mambo

Translated by www.deepl.com(translate

An instrument definition file is a text file that has the extension .ins. The file contains instrument definitions for one or more different instruments. Each can be a specific model of MIDI synthesizer: a keyboard, a portatone, a sound module or a sound card.

See the Cakewalk/Sonar User Guide for instructions on how to create instrument definitions using Cakewalk's dedicated functions.

I prefer a different procedure. If you look at the contents of the .ins file with Notepad or any other text program working in text mode, you can see that the texts are entered clearly structured. With an existing .ins file as a template and the lists of the keyboard documentation (at Yamaha the datalists), it is not difficult to create a new instrument definition. But it costs a lot of careful writing.

In the following I will describe the structure of .ins files, so that you are able to understand the lines of text. I refer to the Tyros instrument definition file tyros_kbds.ins from my homepage, which can be opened with a text program.

This .ins file currently contains instrument definitions for the four Tyros models Tyros, Tyros 2, Tyros 3 and Tyros 4.

The file starts with a block of comment lines from the authors of this file. Comments are introduced by semicolon ";" and apply to the rest of the line. Comments are read over by the sequencers Cakewalk/Sonar and PSRUTI.

.ins files contain six sections in a given order, each starting with fixed identifiers.

1. ".Patch Names" contains text blocks of the voice names
2. ".Note Names" contains text blocks of the note names of drum kits
3. ".Controller Names" contains the names of the controller events
4. ".RPN Names" contains the names of the RPN events
5. ".NRPN Names" contains the names of the NRPN events
6. ".Instrument Definitions" contains text blocks with instructions for the individual instrument definitions.

Each text block begins with an identifier defined by the programmer, enclosed in square brackets. The text blocks of sections 1 to 5 have a uniform structure.

Important:

Each of the instructions explained below has the structure
xxx=yyy,
xxx[n]=yyy or
xxx[n,m]=yyy

Where xxx is a given keyword or even a number. n and m are numbers, occasionally the character "*". yyy is a text block identifier that may contain spaces, but must end with a printable character. Apart from this identifier, statements may not contain spaces. Each statement begins left-justified in a separate line.

Examples:

Richtig: "2=E. Grand Piano {XG}" Wrong: "2 = E. Grand Piano {XG} "
Right: "Key[16128,0]=SFX KitSFX Kit1" Wrong: " Key[16128,0]=SFX KitSFX Kit1"

1. ".Patch Names"

For each so-called voicebank of all instrument definitions realized in the .ins file there is a separate text block.

A voicebank is the group of up to 128 voices defined by MSB (Controller 0) and LSB (Controller 32). The voicebanks are identified by numbers that are defined by the formula "128 times MSB plus LSB".

This is an excerpt from the first text block of the ".Patch Names" section:

```
=============================================
[GM1 & XG Bank 0]
0=Aco Grand Piano {XG}
1=Bright Aco Piano {XG}
2=E. Grand piano {XG}
.
.
.
22=Harmonica {XG}                (Accordion)
.
.
.
124=Telephone Ring {XG}          (Sfx)
125=Helicopter {XG}              (Sfx)
126=Applause {XG}                (Sfx)
127=Gunshot {XG}                 (Sfx)
=============================================
```

With each line, a voice number is assigned the voice name used on the keyboard. Voice numbers (also called patches or program numbers) count from 0 to 127 in Cakewalk. Lines for voice numbers that are not realized by voices in the keyboard are omitted. {XG} or (Accordion) are additions that make it easier to find voices with the browser of Cakewalk or PSRUTI. They are introduced here with a series of blanks, but these belong to the identifier.

With the "BasedOn=Textblock identifier" command you can switch to another text block if the text block does not contain any voice numbers. In the present .ins file BasedOn is only used for drum kits.

2. ".Note Names"

This section contains the drum names assigned to note numbers for each individual drum and SFX kit.

Example:

```
=============================================
[DrumsHitKit]
BasedOn=Standard Kit 1
31=Snare Electro
33=Kick Tight L
.
.
.
45=Hybrid Tom 3
46=Hi-Hat Open 2
47=Hybrid Tom 4
48=Hybrid Tom 5
50=Hybrid Tom 6
54=Tambourine Light
=============================================
```

In the text block, for example, the note value 45 is assigned the name "Hybrid Tom 3". The BasedOn line refers to another text block called "Standard Kit 1".

```
================================================
[Standard Kit 1]
13=Surdo Mute
14=Surdo Open
.
.
.
81=Triangle Open
82=Shaker
83=Jingle Bell
84=Bell Tree
================================================
```

This causes, for example, the note value 82 of the drum kit "DrumsHitKit" to be assigned to the text block "Standard Kit 1"; the drum identifier is thus "Shaker".

3. ".Controller Names"

In this section, each number of the controllers implemented in the keyboard is assigned the corresponding controller name.

This is a section of the text block [Tyros Controllers]:

```
================================================
[Tyros Controller]
0=0 Bank Select MSB
1=1 modulation
5=5 Portamento Time
6=6 Data Entry MSB
7=7 Main Volume
.
.
.
91=91 Reverb Send Level
.
.
.
126=126 Mono
127=127 Poly
================================================
```

Since all Tyros keyboards use the same controller names, there is only one text block.

4. ".RPN Names"

In this section, each number of the RPN instructions implemented in the keyboard is assigned the corresponding identifier. Again, there is only one text block.

```
================================================
[Standard RPN]
0=Pitch Band Sensitivity
1=Fine Tune
2=Coarse Tune
5=Modulation Sensitivity (GM2)
```

```
16383=Reset RPN (zero)
================================================
```

The RPN number is calculated from the values of controllers 100 and 101 according to the formula "Ctrl101 times 128 plus Ctrl100".

5. ".NRPN Names"

Again, there is only one block of text.

```
================================================
[NRPN Tyros]
0=VH Mute
136=Vibrato rate
137=Vibrato Depth
138=Vibrato Delay
154=VH Detune Modulation
160=Filter Cutoff Frequency
.
.
.
6819=Drum EQ Treble Freq. F10
6820=Drum EQ Treble Freq. F#10
6821=Drum EQ Treble Freq. G10
================================================
```

The NRPN number is calculated from the values of controllers 98 and 99 using the formula "Ctrl-99 times 128 plus Ctrl-98".

6. ".Instrument definitions"

For each of the .ins file realized keyboard models there is a special text block. The first block is introduced by the identifier [Tyros]; the others by [Tyros2], [Tyros3] and [Tyros4]. All these text blocks have a uniform structure.

We will deal here exclusively with the example of Tyros 4.

```
================================================
[Tyros4]
Control=Tyros Controllers
RPN=Standard RPN
NRPN=NRPN Tyros

Patch[0]=GM1 & XG Bank 0
Patch[1]=XG Bank 1 (KSP)
.
.
.
Patch[115]=Tyros4 Bank 115
.
.
.
Patch[16128]=Drums+SFX Tyros4
Patch[16256]=Drum Kits Tyros4
Patch[*]=GM1 & XG Bank 0

Key[15360,0]=GM2 Standard
.
```

```
.
.
Key[16128,0]=SFX KitSFX Kit1
.
.
.
Key[16256,89]=Live!DrumsAcousticKit
Key[16256,90]=Live!DrumsRockKit
Key[16256,91]=Live!DrumsRealDrums

Drum [15360,*]=1
Drum [16256,*]=1
Drum [16128,*]=1
==============================================
```

The lines begin with the keywords

(a) "Control"
(b) 'RPN
(c) 'NRPN
d) "Patch"
e) "Key"
f) "Drum"

With the exception of "Drum", the equal signs are followed by identifiers of text blocks in sections 1 - 5.

a) Control=Tyros Controllers

Effect: The names of the controllers are defined in the text block "Tyros Controllers".

b) RPN=standard RPN

Effect: The names of the RPN are defined in the text block "Standard RPN".

c) NRPN=NRPN Tyros

Impact: The names of the NRPN are defined in the text block "NRPN Tyros".

d) The patch instructions

Example "Patch[115]=Tyros4 Bank 115".

115 is the number of a voicebank here; "Tyros4 Bank 115" is the identifier of the assigned text block in the ".Patch Names" section.

Effect on the example MSB = 0, LSB = 115, (=Voicebank 115), voice number = 21:

The voicename results in "SmallAccordion {T}   (accordion)"

The last patch instruction "Patch[*]=GM1 & XG Bank 0" causes voicebanks not covered by patch instructions to take the voicenames from the text block "GM1 & XG Bank 0".

e) The key instructions

Example   Key[16128,0]=SFX KitSFX Kit1

16128 is the voicebank defined by MSB = 126, LSB = 0 Yamaha keyboards are the bank of SFX kits. The second parameter 0 of the Key instruction is the voice number.

The statement thus causes the note names to be defined in the text block "SFX KitSFX Kit1". The note value 52 therefore includes the sound "Flute Key Click".

d) The drum instructions

These instructions determine whether all or individual voices in a voicebank are melody voices or drum or SFX kits. In contrast to melody voices, drum/SFX kits generate different drum instruments or sounds with each keyboard key.

Example:

Drum [16128,*]=1

This instruction sets all voices of bank 15360 (MSB/LSB=120/0) as drum/SFX voices.

For example, drum [1234,34]=0 or 1 could be used to set the property of a specific voice 34. All voices of banks for which no drum instruction exists are melody voices by default.

Create a new instrument definition: The procedure in practice

First you have to decide whether you want to extend an existing .ins file or create a separate .ins file for the new instrument definition.

Especially if you are dealing with similar keyboards from one manufacturer, it is advisable to extend an existing file. The reason is that many existing sections (e.g. .Controller or .NRPN) and also text blocks can be included. For a new .ins file you have to copy all these sections.

As a first step, it is recommended to use the text block for the new keyboard model in the ". Instrument Definitions" section, preferably by copying an existing one. After entering the keyboard name, **all** instructions must be checked step by step to ensure that the contents of the corresponding text blocks in sections 1 to 5 also apply one hundred percent to the new keyboard. If this is not the case, e.g. if new voices have been added or no longer apply, additional text blocks with new meaningful identifiers must be created in the relevant sections and carefully filled in using the keyboard documentation (in Yamaha's case the Datalist). The references in the relevant instructions must be adjusted accordingly. It is essential that the identifiers match the text block identifiers exactly. Under no circumstances may the existing text blocks be changed. For new voicebanks new patch instructions must, of course, also be inserted. Patch instructions must be deleted if the voicebank is not supported by the keyboard.

Checking the entries against the keyboard documentation and entering new voice or note names is very time-consuming; especially since this work has to be done very carefully.

From my homepage http://www.heikoplate.de/mambo you can download a freeware program under Downloads - InsFileChecker, which can be used to check the consistency of the identifiers of the text blocks and the instructions. The program checks whether at least one reference in the section ".Instrument Definitions" exists for each text block and whether there is a corresponding text block for each reference.

A very compact but complete format description of Cakewalk .ins files was written by Pete Goodliffe: Cakewalk .ins files
I refer here in particular to the "BankSelMethod" instruction, which I have not described in this paper, as it is not required for Yamaha keyboards.

End