midifan月刊

第06期

免费技术月刊 —— 我们关注电脑音乐

Guitar Rig 2 VS. AmpliTube 2
模块化吉他效果器软件的对决

09月号
总第06期
2006年09月10日发行



MidAir25 无线MIDI键盘



体验SSL魅力

声卡葵花宝典

骆驼加蜥蜴的神秘之声

浅谈流行歌曲的编曲

我们需要什么样的分轨

如何玩好音高、音量、音色、音长



免费下载

主编：泉斓
策划：musiXboy
www.midifan.com

众多小贴士：水果N连音神功、你也许不知道的Pro Tools实用快捷键

Reason：正确调节鼓loop的音高、Cubase SX：用外部工具直接快速导出分轨……

midifan月刊

关于我们



泉斓

《midifan月刊》主编，负责稿件的编排与版面设计。◀



musiXboy ◀

《midifan月刊》策划，负责月刊专题的选材与稿件的审核。

特约作者简介



张火 网名“大觉者”，MIDIFAN论坛总版主。职业音乐制作人、录音师。毕业于山东艺术学院音乐系作曲专业，大学时期开始学习电脑音乐并创办音乐工作室。现任职山东省艺术馆文艺部，担任音乐制作及录音工作。◀



门子 名叫萌克 MIDIFAN论坛总版主，制作人。钢琴学了十年连一级都没过，接触midi后不能自拔，拜了不少师父，学了点手艺，也算是混口饭吃。目前自己单干中。◀





艺名：小康　网络ID：康师傅 ▲

淦立康 1974年出生，音乐教育专业毕业，组建过多个乐队，现任厦门集美大学艺术教育学院音乐系数字音乐创编专业的专业讲师，是中国第二代数字音乐网络推广人之一；拥有多年的数字音乐制作和音频处理经验。

熊猫 男 23岁 北京人

喜欢睡觉

专业球迷

Java程序员

硬件爱好者

DIY发烧友 ▲



卢小旭 音乐制作人。

长期担任“水木年华”专辑制作人，为其编曲。制作了大量的歌曲，为张娜拉《功夫》、孙悦《幸福沙漏》专辑创作多首歌曲。编撰出版了《Cubase&Nuendo电脑音乐制作》《电脑音乐轻松入门》《电脑音乐技术与实践》《MIDI技巧与数字音频》多部书籍。▶

特约作者简介



Pediatrician

自学英语，电脑。从事儿科专业、影视后期制作。看到新手的学习欲望很高，于是充个数，做点易学易懂的视频教材，与大家共同提高。如果因为我的嗓子（尚能发音）不够"赵忠祥"而影响你的学习，我只能表示遗憾:)！◀



妹妹

论坛ID妹妹，原名孙玮。
自幼学习钢琴及音乐理论。自16岁起开始接触音序器和合成器，后考入中国戏曲学院MIDI作曲专业。在交流，尝试和实践中逐步提高。曾参与多部动画片音乐及少儿歌曲音乐的创作工作。擅长数字音频原理、音响技术、合成器和采样器原理。现就读于中国戏曲学院MIDI作曲专业的硕士研究生。▲

吴崧铭 ▶

(Bio
(Name March_Wu)
(Occupation Game_Composer)
(pause "大家好，我是新人，请多多关照。
我会聊天，会打字，我还会截图呢！")
);;end of Bio



王逸驰

男，26岁。▶
短歌行成员，擅长数字音频原理、音响技术、DSP原理、合成器和采样器原理。部分作品包括有短歌行朝露系列SoundFont音色库、短歌行"传奇"模拟合成器和FM合成器、千年鹰采样器、中国传统民族乐器采样——独奏二胡、琵琶等。





曾冠宇

justbecrazy

音效设计师
音乐制作人
▼



csounding

其实和csound无关，来源于continuously-sound-ing，估计可以解释成持续不断的发声，迎合了本人对声音的追求不止，但是还不是发烧友，属于依靠科学依据的理智人员，目前隐居中…… ▲



Seer Xillion

真名由于太不好听（个人觉得），故不便告知。现在在一家软件公司做工程师，但一直热爱MIDI。可惜一直以来做的MIDI都听上去太像MIDI，所以在一直不断寻求做得更像商业CD的方法。▲

Bretown佰唐

www.baitang.com.cn

中国数码音乐/音频设备专业旗舰店

即将开业

Coming soon ...



地址：北京市海淀区远大路1号，金源时代购物中心二期二层，音响盛地A1B2室

中音公司

自作主张

玩音乐—我自做主张



澎湃一粒子监听！

精准、纯净、细腻，
大动态、大功率，大箱体
频响平直，低频出色，艺术化设计，
新一代音乐录音标准监听的代表。

CME

· 领 · 先 · 一 · 步 ·

SAC

资 深 音 频 顾 问

中音公司 北京:010--85801115 广州:020--81338963 香港:852--24247368 www.centrmus.com

CONTENTS

点击文章标题 发表评论 与作者互动

主编：泉斓　策划：musiXboy
2006年09月10日发行　总第06期

独 门 秘 籍

8　justbecrazy　实用压缩

11　吴崧铭　CAL 编程从入门到精通（下）
——菜单语句简介

22　卢小旭　骆驼加蜥蜴的神秘之声
——Cameleon5000介绍

27　妹妹　浅谈流行歌曲的编曲（一）：
抒情流行歌曲中的String弦乐组

29　王逸驰　声卡葵花宝典（一）：
声卡接口面面观

33　门子　门子缩混系列连载：在笼子里跳舞
（二）——我们需要什么样的分轨

39　SeerXillion　专业音乐软件开发入门（五）
——解读XSynth（上）

46　csounding　从声音到声音——音色合成技巧（二）
如何玩好音高、音量、音色、音长

抢 先 评 测

58　musiXboy　把USB线和电源线咔嚓掉
——M-Audio MidAir25 无线MIDI键盘评测

62　大觉者　体验SSL魅力
——SSL 4000 Collection效果器插件

对 决 PK 台

67　门子　Guitar Rig 2 VS. AmpliTube 2
模块化吉他效果器软件的对决

今天，托midifan老大的福，门子同时拿到了Guitar Rig2和AmpliTube 2。其中Guitar Rig 2是我买的正版，AmpliTube2是演示版（我是NI迷，所以掏钱买了Guitar Rig2）。看起来用一个正式版和演示版比不是很公平。不过好在AmpliTube 2的演示版在声音上和正式版并没有什么区别，而只是在模块上加入了一些限制。所以在下文的评测中，门子会尽量使用两个软件所共有的模块来进行对比。









视 频 教 程

74 Pediatrician Adobe Audition 2.0 视频面对面（十一）——优化声音波形

74 Pediatrician Adobe Audition 2.0 视频面对面（十二）——增加声音特效

小 贴 士

75 Wode 水果n连音神功

由于水果没有比较方便的做多连音的工具，所以我们在这里给大家讲解一下可以做出n连音的方法，以及提高多连音制作效率的秘诀——n连音神功。

80 musiXboy 用外部工具直接快速导出分轨

我敢说所有Cubase SX的使用者都用的上今天我们要讲的小技巧，这可以节省大量的时间和体力。

Cubase SX几乎可以说是世界上最好用的编曲软件，但是……有一件很重要的事情它做不了，直到Cubase SX升级到3.1版本也无法实现，那就是直接导出分轨。

在Cubase SX导出分轨确实是一件头疼的事情……

83 musiXboy 历史车轮：Native Instruments 发展历史回顾

85 淦立康 超级人声处理机——Sonar V-Vocal （二）

89 musiXboy 你也许不知道的实用快捷键

这一讲，我们来说一说Pro Tools的快捷键。也许你会说那些快捷键我早就在用了，我当然不会照着说明书一个一个告诉你这些快捷键，而是要告诉你一些说明书上没有写的、隐藏的很深的、却又非常有用的快捷键。

91 musiXboy Mac Pro和Mac OS X 10.5 Leopard

2006年8月7号，苹果在WWDC大会上，宣布了两件重头产品：Mac Pro 64位桌面电脑，以及64位内核的Mac OS X 10.5 Leopard（美洲豹）操作系统。

93 熊猫 PC电脑趋势谈——扣肉全接触

96 musiXboy 正确调节鼓loop的音高

一直不能让Reason的鼓节奏与你的旋律很好的结合在一起吗？如果你从来没考虑调节鼓采样或鼓loop的音高，来创造与旋律更融合的混音，那么你就要仔细看看这篇文章了。



P80



P96

本刊声明

★本刊图文版权所有，未经同意不得转载或摘编。

★本刊作者发表的文章仅代表作者个人观点，与本刊立场无关。

★向本刊投稿者不得一稿多投。本刊对来稿有删节权，不同意删改者请注明。

★来稿必须保证是作者原创，如有抄袭剽窃等侵权问题，本刊概不负责。

★来稿一经采用，稿费从优，如果您对本刊有任何疑问，请将邮件发送至mx@midifan.com 进行咨询。

midifan.com 版权所有©

KORG

中国民乐风格融入PA系列

适用于PA50、PA80、PA1X及PA1X Pro伴奏机，有关免费索取民乐风格的磁盘，请联络雅登各地销售办事处及全国各通利琴行门店。欢迎浏览www.actonchina.com下载PA系列的“.SET”应用档案和MP3格式来试听中国民乐风格。

中国总经销

Acton
雅登音响乐器(上海)有限公司

中国总部
地址：上海市浦东新区
唐镇王港虹叶路3号
电话：86-21-5858 9280
传真：86-21-5868 0531
网站：www.actonchina.com
电邮：enquiry@acton-sh.com

全国办事处
上海办事处
电话：86-21-5858 9280
传真：86-21-5868 0531
北京办事处
电话：86-10-67020066
传真：86-10-67019993

广州办事处
电话：86-20-8339 7731
传真：86-20-8339 7428
成都办事处
电话：86-28-8522 7631
传真：86-28-8522 7632

TOM LEE Music 通利琴行

通利琴行北京旗舰店
地址：北京市崇文区天坛路57号
电话：010-67020099

深圳：(755) 8329 3828　上海：(021) 5228 2065
广州：(020) 8386 4038　宁波：(574) 8731 3256

中音公司

自作主张

玩音乐—我自做主张



U-Key‘魔琴’——让你5分钟学会演奏自己喜欢的乐曲！

从来没有一款键盘象U-Key‘魔琴’这样特别：超薄电脑音乐主键盘，‘拍歌’即时更新曲库，演奏技巧测试功能，多国音阶智能演奏，全MIDI音乐软件控制……‘魔琴’，你桌面上的精灵！

CME

领 · 先 · 一 · 步

SAC

资 深 音 频 顾 问

中音公司 北京:010--85801115 广州:020--81338963 香港:852--24247368 www.centrmus.com

点这里进入论坛讨论此文

实用压缩

Justbecrazy/文

这是继6月那篇所谓的“实用混响调节技巧”之后的一篇关于实用压缩技巧的文章。既然谈到“实用”，我又很怕各位朋友把“实用”这个词绝对化了，认为就应该是这样。其实像上次一样，我要强调的，还是“耳朵第一”原则。这篇文章我们会先介绍传统的全频段压缩器的使用，然后再深入到多段压缩和母带制作过程的压缩。

压缩的两个用途

先要明确的一个事情是，如果你要把一个压缩器的所有参数都调正确，那么首先你要知道的是，你要取得什么样的效果。我比较喜欢简化事情，所以这里我把我们要用压缩器做到的用途简化成下面两个：

第一个用途是控制峰值。所以你如果想把峰值降下来，然后又不会影响到其他部分，经常用的一个办法是把threshold设在刚好平均电平的上方，这样做的好处是只有峰才会被压到。如果你想让这些峰值被压得多点，就把ratio设大点，一般在这种用途下，ratio设在2.5:1到8:1之间。

在有些情况下，用高ratio，短attack和短release来决定threshold会更加方便，因为这种设定下可以很清楚地从gain-reduction表里看出峰值被压下了多少。设定时，你可以把threshold慢慢往下降，一边看着表，如果波形上大峰出现时，表有明显的下降，而在两个大峰之间，表没有明显的读数，那么这个threshold就可以满足这第一个用途。在确定了threshold之后，就可以反过来确定attack和release，然后再来确定ratio。一个比较常用来确定ratio的方法是看gain-reduction表，一般认为最大的电平下降值在8dB到10dB。说到这里我又想说“耳朵第一”原则了，毕竟表中的数值只是作为一个参考，你要仔细听这种声音是不是你想要的。如果声音已经被压成一个大饼，那么你就要把ratio设低，或者把attack设长。另外，如果你想对峰值有着严格的控制，那么一个hard knee的压缩器将有更好的效果。而且，一个有着把峰值作为检测对象（还有一种是把RMS作为检测对象，下面会提到）的side-chain模式压缩器对峰值的监控会更加准确。

即使你用压缩器来控制峰值，你也要注意这一点：如果你不是用一个很狠的压缩器，而且又没有设定一个很快的attack的话，肯定会有一些快峰被你漏过去的。为了避免这种情况的发生，可以在压缩器的后面串一个限制器。这种做法在母带制作时十分常见，毕竟单单靠一个压缩器就想防止电平劈，是不大可能的事情。

第二个用途是压下整个动态，而不只是压下峰值。如果是这样的用途，那么通常的做法是设一个很低的ratio，一般在1.1:1到1.4:1之间，然后把threshold设定峰值之下30dB的地方。Soft-knee的压缩器在这里挺好用的。RMS的作为检测对象也比峰值用得多。

多段压缩与母带制作的压缩

这里有一个经常问的问题是：“为什么我在mix的时候，甚至在录音前期，已经对各个轨进行压缩，为什么还要在总输出上再压一次？”答案是，其实并不是所有的音频都需要压缩，但在总体上来一个压缩可以让

你的mix更实更有力度，即使是在每一个单轨都已被压得很扁的情况下。因为，每一轨都压得很扁并不表示总输出上也是如此。比如，你的人声在两句之间总有空隙，还有乐器在很多时候也不是从头弹到尾吧，所以如果没有进行总体的压缩，那么最后输出结果动态波动依然很大。

既然是动态变化的可能性比较大了，所以用一个带有自动设定attack和release的压缩器进行压缩，可能会比较简单。但如果没有，可以试试把attack设定在大约20ms，release设定在大约300ms，当然还是要以“耳朵第一”作为第一原则。因为压缩器都有自己的“性格”，不同的压缩器出来的声音都不一样。用一个低theshold＋低ratio加在mix上，你会瞬间感觉你的mix更有力了，而且响度提升了。但如果压得太狠，这个时候传说中的“pumping”就会发生。但如果提升量往回减小2到3dB，那么主观上的听觉结果将会挺爽，而且对突出主要部分很有帮助，比如人声之类。

当你把一个传统的压缩器加在整个mix上时，很容易发生“pumping”，这主要是因为高能量的低频在起着主要作用。多段压缩在这种情况下就可以用，因为你可以对不同的频段采用不同的压缩方法。在多段压缩中，音频按频段被分成几个独立的部分（一般是三个），然后每个部分分别经过各自的压缩器，然后在输出的时候，各个部分再合在一起，形成一全频段的音频信号。

多段压缩一个很明显的优势是，高能量的低频在被压缩的同时，中频和高频却不受影响。这一点上多段压缩与传统的全频段压缩是相反的：在以前，压下一个大声的底鼓，就会同时压下同时在响的踩镲和军鼓，而后面这两个你可能不压都嫌太小声呢。总之，对声音进行更大的压缩，却产生更小的副作用，是多段压缩的主要好处。

另一方面，一个传统的全频段压缩在变成多段压缩之后，自然可以调节的参数也就更多。比如每个频段都有一个增益，由于这就相当于EQ的作用，所以改变这个参数就会改变音色。比如你在压缩了之后觉得中频稍少了，可以简单地提高一下中频段的增益，很方便。如果你压下了低频之后，又对低频增益进行提升，那么低频的整体响度就提升，但是低频的峰值却没有提升（因为你压过了），所以你的低频听起来就更加有力。同样的道理，如果你要增加一点高频空气感，也可以采用类似的方法：对高频进行压缩，然后对高频增益进行提升。

很多多段压缩都可以允许你去移动分频点，一般的三段压缩，它的两个分频点是把频率分成：高，中，低音三个频段。一般低频段要包括底鼓，贝司这类乐器；中频比较宽，一般要包括人声的整个频段（除了一些高频泛音和气流声）；高频段要包括镲片，木吉他扫弦声等。分频点的位置也很重要，你可以想想你把一个分频点刚好卡在人声频段的中间，你在做多段压缩的时候是多么麻烦。对于经典的流行歌来说，常见的两个分频点是：150到200Hz，5到8Hz。

如果是其他风格的东西，你要仔细听，分出你的mix中的几个频段，然后才能设置相应的分频点。分频点设置完后，就是对每一段进行压缩处理。这一步非常需要经验，但首先你要问自己：这仅仅是个频率不均的问题？还是某个频段的动态需要调整了呢？一旦找到了问题，就争取用最少的步骤的解决它。

在母带制作的过程中，采用多段压缩确实可以解决很多问题。一个常用的方法是在低频段采用高ratio和高theshold，在其他频段采用低ratio和低theshold。另一方面，在中频段进行压缩有利于突出人声。

还有一点，在母带制作过程进行多段压缩时，如果你没有很明确的理由，那么对不同的频段经常采用相近的attack和

release。Attack要求在听不出很突兀的压缩感觉的前提下尽量快点，但有些情况下，比如你要更突出你mix中动态瞬间改变带来的力度感，你就要把attack稍微放长点。如果你的attack和release设置相差太多，那么可能会发生：在某个时刻，某个频段的声音早于其他频段出现；或者出现的力度过强了。如果极端下去，甚至可能会发生：你听起来各乐器之间根本不在同一个拍子上，很恐怖。

最后

相对于其他效果，比如延时，混响等，压缩产生的变化是更加细微的。这就意味着，你要玩好它，可能需要更多的时间和经验。事实上，对动态的控制，是音乐制作，尤其是现代的音乐制作中一个非常重要的主题，不管是什么风格。而且要一直记住，不同的压缩器会产生不同“性格”的声音。另外，多看看人家压缩器的预设，多想想，为什么人家的这个预设要采用这些参数，在这些参数下，声音听起来会是怎么样的，从这些东西中，你会学到很多。再有就是多听市面上的一些CD了。最后，学会克制，宁缺勿滥，不管什么效果，不管是不是压缩，你加过了造成的危害，要远比加少了来的多！谢谢！

参考文献：Paul White，ADVANCED COMPRESSION TECHNIQUES，SOS，2001.1.



CAL编程从入门到精通(下)
菜单语句简介

吴崧铭/文

上期程序代码样例：

练习四、把这个数学式编一个程序来计算结果： (465+532/2-799)*11+(87-53)/2

```
(+ (- (+ 465 (/ 532 2)) 799) (/ (- 87 53) 2) )
```

练习五、编一个程序来计算4*3*2*1的值(此练习相关的文件为train-05.cal)

```
(do
 (int int_counter 1)
 (int int_result 1)
       (while (<= int_counter 4)
         (do
           (*= int_result int_counter)
             (++ int_counter)
         );; end do
       );; end while
  (pause "the result is" int_result)
) ;; end do
```

练习六、编一个程序使音轨里所有音符的力度值量化成5的整数倍(此练习相关的文件为train-06.mid train-06.cal)

```
(do
 (forEachEvent
  (do
   (if (== Event.Kind NOTE) ;;判断事件是否为音符事件，如果不是音符事件略过
    (do
     (/= Note.Vel 5) ;;对力度值整除5
      (if (== Note.Vel 0);;如果整除后值为0
       (= Note.Vel 5)  ;;则对力度赋值为5
       (*= Note.Vel 5) ;; 否则对整除后的值乘5
        );; end if
      );; end do
     );; end if
    );; end do
  );; end for
);; end program
```

练习七、把所有的使用7号控制器控制的变化信息改成11号控制器来实现。

使用CW的Interpolate替换功能既能实现，不用设计CAL程序;)

练习八、删除所有不匹配的延音踏板信息，当我们使用踏板录制钢琴演奏时，往往出现踩着踏板停止或者踩着踏板开始的现象，这时会造成某些地方踏板开被录下来但却没有关闭信息，或者在关闭踏板信息之前没有踏板开信息。通过这个程序使所有踏板信息都是一开一关配对出现，并最后结束在关闭上。(此练习相关的文件为train-08.cal train-08.mid)

程序应该对64号控制器作判断，可以设定一个变量来存储当前64号控制器的值，当64打开时置该变量为127，当64关闭时置为0，当碰到下一个64号控制器时，将该变量的值与当前的64号事件作比较，当前64号控制器与该变量的值不能相同，如果相同则在当前控制器前10个tick处插入一个相反状态的值。全曲完后还应该使64号控制器置0。程序如下：

```
(do
 (int int_lastval 0);;开始时将上次踏板值假设为关闭状态
  (do
   (forEachEvent
```

```
    (do
     (if (&& (== Event.Kind CONTROL)
(== Control.Num 64))
       (do
        (if (== int_lastval Control.
Val);;如果上一个控制器与当前状态相同
          (do
           (if (== Control.Val 0)
            (insert (- Event.Time 10)
Event.Chan CONTROL 64 127) ;;如果当前64
号为0则插入一个打开事件
             (insert (- Event.Time 2)
Event.Chan CONTROL 64 0) ;;如果当前64号
不为0则插入一个关闭事件
            )
           )
          );; end if
        (= int_lastval Control.Val) ;;将
当前控制器状态存入变量中准备下一次判断
         );; end do
      );; end if
    );;end do
   );; end for
    (if (== int_lastval 127) (insert (+
Event.Time 2) Event.Chan CONTROL 64 0))
    ;;如果最后一个64号不为0则插入一个关
闭事件
  );; end do
);; end program
```

练习九、根据用户输入对弯音值作缩放处理，有时我们发现当我们把弯音范围放大到一个八度时，原来有些段落按大二度作的弯音就变得极为恐怖，这个程序让用户根据需要对弯音值按比例缩小或放大。

使用interpolatet替换功能比编一个CAL要简单得多，我们毕竟不是为了编程序而编程序，我们是要为我们的创作提供更方便更快捷的手段，所以能够不用CAL的尽量不要用，能用现成的CAL时尽量不要自己编，能改改别人的CAL就用时尽量不要重新编，实在不行了再来自己编吧！

不过通过对练习七和练习九的练习相信您现在对使用CAL编程已经很熟悉了，下面的内容我们介绍一下通过CAL控制系统菜单。

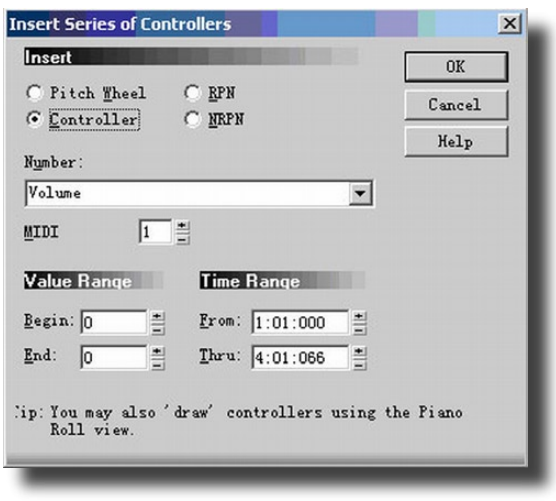
菜单控件语句

CAL提供语句对CW和SR的系统菜单中的一些功能进行调用，这些语句大多与菜单操作具有相同的功能，下文将这些语句罗列一下，并与相应的菜单项对应说明，除了特别之处外其他将不作详细介绍了。需要特别注意的是，不同版本的菜单内容和结构也不相同，有些虽然名称相同但部分功能也不相同，下文有些语句带有40标记，表示只有CAL4.0版才支持的语句，请注意自己所使用的版本。

Edit菜单

Edit菜单中包含许多我们经常使用的功能，而这些功能事实上都可以用CAL编程来实现，然而，笔者并不推荐大家动不动就用CAL来编程，巧妙地运用CW及SR已有的工具很多时候足以应付大多数工作的需要。而CAL语句里面提供的对系统菜单的调用从某种意义上说其实有些多余。

EditControlFill 在程序中输入此语句将调出一个填充事件的窗口，如下图：



用户可以自己输入相关值实现事件填充，可以对指定的时间段内特定事件进行平均的变化填充，例如插入7号控制器渐变效果，大家可以自己试一试就知道效果如何了。该语句也提供参数直接在CAL里进行相应的填充控制，这时将不会弹出此窗口，该语句详细描述为：

(EditControlFill [<开始时间> <结束时间> <控制器号> <通道号> <起始值> <结束值>])

这时无法实现对WHEEL、RPN、NRPN事件的填充。

EditCopy对应edit菜单的copy，语句结构：

(EditCopy [<开始时间> <结束时间> <事件> <过滤器> <速度> <拍号调号> <记号>])

不提供任何操作数时，该语句将打开CW的copy对话框，操作数中事件、过滤、数度、拍号调号、记号都是布尔型值，只能取TRUE/FALSE，TRUE表示此次复制的对象包括此类数据，FALSE表示不复制此类数据。

EditCopy40与EditCopy相同，语句结构略有不同：

(EditCopy40 [<事件> <速度> <拍号调号> <记号> <音频> <块关联>])，适合4.0以上的版本。

EditCut对应Edit菜单的Cut，语句：

(EditCut [<开始时间> <结束时间> <事件> <过滤器> <速度> <拍号调号> <记号> <删除空>])，不提供任何操作数时将弹出剪切对话框。

EditCut40与上面相同，语句结构：

(EditCut40 [<事件> <速度> <拍号调号> <记号> <删除空> <分块信息> <小节对齐>])

EditDelete删除事件，注意别把此删除与(delete)语句搞混了，这里是调用Edit菜单里的删除操作，语句格式如下：

(EditCut40 [<事件> <速度> <拍号调号> <记号> <删除空> <分块信息> <小节对齐>])

如果没有任何操作数，该语句将弹出删除对话框。

EditFitImprov速度聚合功能，与edit里的fit improvisation相同。其功能是自动调整系统速度，使即兴演奏的几个音吻合到拍子上。这一功能相信大家都用过，不过似乎没有CW所标榜的那么好。语句：

(EditFitImprov [<音轨号>])

EditFitImprov40同上，该语句格式省略<音轨号>:

(EditFitImprov40)

EditFitToTime速度聚合到时间，同Edit菜单的Fit to time功能，语句如下：

(EditFitToTime [<开始时间> <结束时间> <聚合后的结束时间> <聚合方式> <聚合音频>])

其中聚合方式是一个布尔值，取TRUE时表示修改事件时间值，取FALSE时表示修改系统速度；聚合音频也是一个布尔值，取TRUE时表示聚合音频，FALSE表示不聚合音频。在SR环境里使用此语句会报错“invalid value”，这时建议使用下面的语句。

EditFitToTime40功能同上，语句如下：

(EditFitToTime40 [<聚合后结束时间> <聚合方式> <聚合音频>])

此语句对选中的部份进行聚合，所以省略了开始时间和结束时间两个操作数。

EditGrooveQuantize使用模版量化，同Edit菜单里的Groove quantize。语句：

(EditGrooveQuantize [<开始时间> <结束时间> <是否使用过滤器> <量化精度单位tick> <灵敏度百分比> <开始时间精确度> <时值精确度> <力度精确度> <四种量化移动方式> <模版源>])

其中几个精确度设定值都是百分比，三

种移动方式为0=Do Not Change 1=Quantize to resolution 2=Move to nearest 3=Scale time，模版源为模版文件的路径和文件名，clipboard表示使用剪贴板里的数据作为模版源。

EditGrooveQuantize40与上述功能相同，语句格式为：

(EditGrooveQuantize40 [<量化精度单位tick> <灵敏度百分比> <开始时间精确度> <时值精确度> <力度精确度> <四种量化移动方式> <模版文件名> <模版名> <量化对象>])

其中量化文件名和模版名都是文本值，要用双引号括起来，量化对象为布尔值，TRUE表示歌词、音符和音频数据，FALSE表示仅对音频数据有效。

EditInterpolate Edit菜单里的Interpolate功能，语句如下：

(EditInterpolate [<无对话框>]

该语句只有一个操作数，当忽略时弹出Interpolate对话框，否则将使用该值指定的CAL过滤器设置。

EditLength功能是Edit菜单里的Length功能，语句如下：

(EditLength [<开始时间> <结束时间> <过滤器> <百分比> <对开始时间起作用> <对时值起作用>])

最后两个操作数为布尔值。

EditLength40功能同上，语句如下：

(EditLength40 [<百分比> <开始时间起作用> <对时值起作用>])

EditPaste相当于paste的功能，语句如下：

(EditPaste [<粘贴起始点时间> <循环次数> <3种粘贴模式> <粘贴事件> <粘贴速度变化> <粘贴拍号和调号变化> <粘贴记号信息>])

最后四个操作数均为布尔型，用于指定需要粘贴哪些信息。3种粘贴模式为0=混合粘贴 1=覆盖粘贴 2=插入粘贴

EditPaste40 功能同上，语句格式为：

(EditPaste40 [<对齐开关> <各循环块间是否连接> <what to选项> <目标音轨> <合并音轨> <循环次数> <循环间隔> <是否粘贴为一个块> <事件开关> <速度开关> <拍号调号开关> <记号开关> <3种粘贴模式> <是否分割音频数据>])

EditQuantize量化菜单，语句如下：

(EditQuantize [<起时时间> <结束时间> <过滤器> <精度单位tick.> <是否对开始时间起作用> <是否对时值起作用>])

EditQuantize40同上，语句如下：

(EditQuantize40 [<量化精度单位tick> <量化百分比> <起始时间开关> <时值开关> <摇摆百分比> <量化精度百分比> <偏移量tick值> <2种量化对象开关> <是否缩放音频>])

EditRetrograde与菜单中Retrograde同功能，语句如下：

(EditRetrograde [<起时时间> <结束时间> <过滤器>])

EditRetrograde40功能同上，语句结构如下：

(EditRetrograde40)

EditSlide相当于Edit菜单里的Slide功能：

(EditSlide [<起始时间> <结束时间> <过滤器> <需移动的数量> <单位>])

EditSlider40语句如下：

(EditSlider40 [<需移动的数量> <是否以ticks座单位> <是否移动Events> <是否移动Markers>])

EditTranspose语句：

(EditTranspose [<起始时间> <结束时间> <过滤器> <移动音数> <是否移动全音>])

EditTranspose40语句：

(EditTranspose40 [<音数量> <是否全

音〉〈是否移动音频〉])

EditVelocityScale语句：

(EditVelocityScale [〈起始时间〉〈结束时间〉〈过滤器〉〈起始值〉〈结束值〉〈是否使用百分比〉])

EditVelocityScale40语句：

(EditVelocityScale40 [〈起始值〉〈结束值〉〈是否使用百分比〉])

ResetFilter语句，此语句与接下去介绍的2个语句共同实现过滤器的设定，在下面很多语句中都会包含〈过滤器〉这样的操作数，其中的值就是通过此语句以及下面2个语句实现的对过滤器的设定，设定过滤器主要是指定CAL语句在运行时实施的对象。语句如下：

(ResetFilter 〈类型〉〈全部〉)

〈类型〉的设定为：

0=使用事件过滤器

1=使用Interpolate里的选择过滤器设定。

2=使用Interpolate里的替换过滤器的设定。

〈全部〉=TRUE时表示对所有选中的事件生效。

SetFilterKind语句根据上面语句中〈全部〉的值来设定，如果上面语句中〈全部〉=FALSE，此语句才有效果。

(SetFilterKind 〈类型〉〈事件种类〉〈包含〉)

〈类型〉的设定为：

0=使用事件过滤器

1=使用Interpolate里的选择过滤器设定。

2=使用Interpolate里的替换过滤器的设定。

〈事件种类〉取值有：NOTE, PATCH, CONTROL, CHANAFT, KEYAFT, WHEEL, SYSX, CHORD, NRPN, RPN, EXPRESSION, HAIRPIN, TEXT, WAVE, LYRICS, MCI, WHEEL

〈包含〉为一个布尔值，当为TRUE时表示按前述操作数设定的过滤器选择事件，否则表示前述操作数设定的过滤器之外的事件。

SetFileterRange 〈类型〉〈取值事件〉〈包含〉〈最小值〉〈最大值〉)

〈类型〉的设定为：

0=使用事件过滤器

1=使用Interpolate里的选择过滤器设定。

2=使用Interpolate里的替换过滤器的设定。

〈取值事件〉包括：

0 Note Key

1 Note Velocity

2 Note Duration

3 Key After touch Key

4 Key After touch Pressure

5 Controller Number

6 Controller Value

7 Patch Number

8 Channel After touch Pressure

9 Wheel Amount

10 Channel

11 Beat

12 Tick

13 Patch - Bank Number

〈包含〉是一个布尔值，当取FALSE时表示选择之外的部分。

最小值最大值是指〈取值事件〉的取值范围。

File菜单

文件菜单包含6个可用的语句，实现文件菜单中的相应功能，现将他们罗列如下：

文件保存：

(FileExtract [〈路径〉])

(FileSave)

(FileSaveAs [〈路径〉])

文件打开：

(FileMerge [<路径>])

(FileOpen [<路径>])

文件新建：

(FileNew [<模版名>])

GOTO菜单

(GotoSearch [<noPrompt>]) Go菜单里的Search功能：

(GotoSearchNext)

Setting菜单

(SettingsChannelTable [<bOn> <n1> <n2> <n3> <n4> <n5> <n6> <n7> <n8> <n9> <n10> <n11> <n12> <n13> <n14> <n15> <n16>]) n1-n16为音轨号，取值0-255。

(SettingsMetronome [<bPlay> <bRec> <bAcc> <nCount> <nPort> <nChan> <nKey> <nVel> <wDur> <bBeep>]) 设置节拍器，与节拍器设置窗口里的项相对应。

(SettingsMidiIn [<n1> <n2> <n3> <n4> <n5> <n6> <n7> <n8> <n9> <n10> <n11> <n12> <n13> <n14> <n15> <n16>]) MIDI In设置。

(SettingsMidiOut [<bTxMidiRt> <bSendCont> <bSendSPP> <nSPPDelay> <bCtrlZero> <bCtrlChase>]) MIDI OUT设置。

(SettingsMidiThru [<nMode> <nPort> <nChan> <nKey> <nVel> <nLocalOnPort>]) MIDI Thru设置。

(SettingsRecordFilter [<bNote> <bKeyAft> <bControl> <bPatch> <bChanAft> <bWheel>]) 录制过滤器设置。

Track菜单

(TrackName [<szName> [<nTrack>]]) szName表示音轨名，它可以用来接收其他变量的值从而实现像message显示那样的变化功能，想象一下message里的信息在音轨列表里不断变化显示是怎样的效果。

(TrackActive [<nActive> [<nTrack>]]) nActive取值有1=播放 0=静音 -1=关闭。

(TrackArchive [<nArchive> [<nTrack>]]) nArchive取值有 1=加上存档属性 0=取消存档属性 -1=关闭。

(TrackBank [<nBank> [<nTrack>]]) nBank取值-1-16384。

(TrackChan [<nChan> [<nTrack>]]) nChan取值0-15。

(TrackKey+ [<nAmount> [<nTrack>]]) nAmount取值 -127-127。

(TrackPan [<nPan> [<nTrack>]]) nPan取值 -1-127。

(TrackPatch [<nPatch> [<nTrack>]]) nPatch取值-1-127。

(TrackPort [<nPort> [<nTrack>]]) nPort取值 0-15。

(TrackSelect <nState> <nTrack>) nState取值 1=选中 0=不选中 -1=取消。

(TrackTime+ [<lAmt> [<nTrack>]]) lAmt取值为tick值。

(TrackVel+ [<nAmount> [<nTrack>]]) nAmount 取值 -127-127。

(TrackVolume [<nVolume> [<nTrack>]]) nVolume取值 -1-127。

其他语句

MIDI信息发送语句允许通过CAL直接发送一个midi事件给midi设备，其语句格式如下：

(sendMIDI <端口号> <通道号> <事件类型> [<值1> <值2>...])

端口号和通道号取值-1-15，当取值为-1时向所有端口或所有通道发送。事件类型为事件类型名常量，与前面讲过的insert语句所有支持的事件类型相同，<值1>-<值

N>为每种事件对应的参数值，个数根据不同事件而定。

退出程序语句有：

(error) 此退出返回给CW一个错误代码。

(exit) 退出CAL程序，不返回任何错误代码。

等待语句：

(delay <毫秒值>) 让程序暂停<毫秒值>，1毫秒=1/1000秒。

NIL语句：

NIL，什么都不作，不要括弧，此语句主要用来占一行位置，在最初设计一个程序时，有些结构的语句需要过一会再写，可以用NIL占用一行先，回头再补上。

到目前为止，我们把CAL编程的一些手段和工具都介绍完了，我们在网络上也可以得到很多免费的CAL代码，大家可以从修改别人的代码开始，逐渐地掌握编写CAL程序的方法。在结束此文之前我想再重申我的观点，任何软件或设备，仅仅是我们从事某项工作的一种工具，在音乐制作领域而言，音乐本身才是我们所追求的，熟练使用某几个软件或设备并不代表什么。我们在谋求某种进步和超越的同时往往容易忽视那些我们最初想达到的目的。所以我个人认为，我们在钻研音乐制作技术和设备的同时应该时刻铭记自己最初的打算，专注自己的目标，更多的对自己的音乐理论作更新。

附：一些CAL程序代码及简析(点此下载)

deletdbl.cal 删除重叠音符事件

```
(do
 (int int_curntKey 0);;循环计数器增量
 (int int_lastNote 0);;删掉的音符计数器
  (dword dwd_endTime 0);;上一音符结束结束时间
  (while (< int_curntKey 128)
    (do
    (= dwd_endTime 0)
    (message "Program is running, please wait." int_curntKey);;在状态栏显示信息
     (forEachEvent
      (if (&& (== Event.Kind NOTE) (== Note.Key int_curntKey)) ;;音符事件判断，同样音高判断
       (do
         (if (< Event.Time dwd_endTime) ;;如果上一音的结束位置在当前音符开始处之后
          (do  ;;则
           (delete)  ;;删掉此音符
           (++ int_lastNote) ;;计数器增量
          )
          (do  ;;否则
            (= dwd_endTime (+ Note.Dur Event.Time)) ;;将当前的音符结束时间作为下一次判断的依据
          )
         );; 内层if结束
       )
      NIL
     );; 外层if结束
    );;for结束
   (++ int_curntKey) ;;循环计数器增量
  );; while结束
 )
  (if (== int_lastNote 0) ;;显示报告
   (pause "No doubled/overlapping notes found!")
   (pause int_lastNote " doubled/overlapping notes deleted.")
  )
);;程序结束
```

以前用过一个undouble的，感觉叠音的判断条件不够全面，所以又编了一个。

本程序使用while循环体嵌套一个forEachEvent循环实现双重循环，程序的运行过程是这样的，首先取一个最低音高值存入int_curntKey中，在这个音高值里遍历寻找，凡是某个音符的结束位置在另一个音符中间或超过另一个音符就认为是重叠，并删除，否则将另一个音符的结束时间取出再与后面的音符去比较。所有音符比较完后取第二个音高值来进行寻找，依次将所有128个音高值找遍，程序结束。

红色字体非常重要，第一句是整个程序的核心语句，通过它来进行判断两个音符是否重叠，第二个红句是给循环计数器，也就是音高值增量，没有这一句程序将陷入死循环。

Velo-5.cal 递减和弦音的力度

```
;此程序仅对各音对齐的柱式和弦起作用。
 (do
  (int prevtim 0) ;;前一音开始时间
  (int samtno 0)  ;;和弦音判断器
   (forEachEvent
    (do
      (if (== Event.Time prevtim);;前一音开始时间与当前音开始时间相同则认为是和音
       (do
        (++ samtno) ;;和弦音判断器增量，表示处理第二个和弦音了
        (-= Note.Vel (* samtno 5))
       )
      )
    (if (!= Event.Time prevtim)
     (= samtno 0);; 和弦音判断器清0，表示一个和弦处理完了
    )
    (= prevtim Event.Time)
   )
 )
)
```

本程序可以将和弦各音力度依次按5递减，可以一次选取所有的和弦，而不用一个一个的做，出错可能：如果选取的所有音中有音的力度值太小，使得递减后的值小于0则会报错，例如：3个音的和弦力度都小于15，则递减后最小力度值会为0，解决方法是先用interpolar将力度都量化到15以上再调用此cal处理。

velo-5-10-15.cal

这个cal可以将所有音符的力度量化成5的倍数，例如原来力度是74，量化后变成75，原来力度是78量化后变成80，大于125的力度都量化成125，小于等于4的力度都量化成5。出错可能：没有进行事件类型判断，所以在用鼠标选择的时候只能选择音符(Note)事件，如果选取了非音符事件有可能出错。

```
(do
  (int modv 0)
  (int getintv 0)
   (forEachEvent
     (do
      (if (> Note.Vel 124)
        (do
         (= Note.Vel 125)
        )
      )
      (if (< Note.Vel 125)
        (do
          (= modv (% Note.Vel 10))
       (= getintv (* (/ Note.Vel 10) 10))
        (if (> modv 4)
          (= Note.Vel (+ getintv 10))
        )
        (if (< modv 5)
          (= Note.Vel getintv)
        )
      )
     )
   )
  )
)
```

playmelody.cal

这个程序仅仅是个好玩的玩意儿，起初做它的时候是因为还没买midi键盘，每次改音色总要修改CW里的音色设定后才听到，有了它我就用快捷键来预览音色。运行它可以把选中的旋律播放一遍，不过不是调用CW的播放功能，而是直接向midi端口1发送音符事件信息。这么做的好处是避免CW里设定好的音色号被发送。它不能播放和声。运行时，程序要求输入一个速度倍数值，数值越小速度越快。接着要求输入向哪个midi通道发送，然后开始播放，按Esc键中止播放。

程序用到了sendMIDI语句和delay语句来实现发送数据和延长时值，使用力度值0来关闭音符事件。代码如下：

```
;play a melody
;wrote by March 2000/4/11
(do
 (int int_port 0)
 (int int_timefreq 1)
 (int int_lstKey 0)
 (int int_lstVel 0)
 (dword dwd_lstDur 0)
 (dword dwd_lstTime 0)
  (do
(getInt int_timefreq "input tempo" 1 9)
  (getInt int_port "input port" 0 15)
   (forEachEvent
     (if (== Event.Kind NOTE)
   (do
    (sendMIDI int_port Event.Chan NOTE
int_lstKey int_lstVel)
    (delay (* dwd_lstDur int_timefreq))
    (sendMIDI int_port Event.Chan NOTE
int_lstKey 0)
   (delay (* (- Event.Time (+
dwd_lstTime dwd_lstDur)) int_timefreq))
   (message Note.Key)
   (= int_lstKey Note.Key)
   (= int_lstVel Note.Vel)
   (= dwd_lstDur Note.Dur)
   (= dwd_lstTime Event.Time)
   )
  )
 )
   (sendMIDI 0 0 NOTE int_lstKey
int_lstVel)
   (delay (* dwd_lstDur int_timefreq))
   (sendMIDI 0 0 NOTE int_lstKey 0)
   (delay (* (- Event.Time (+
dwd_lstTime dwd_lstDur)) int_timefreq))
 )
)
```

Rove.cal

这个程序通过10号控制器给音乐添加飘忽不定的感觉，原理为由用户设定一个值，该值为音符的力度限制，小于该力度的音符将获得忽左忽右飘摇的控制，大于该力度的音符其声相被设定在正中。

由于可能出现多音开始时间相同的和声状态，故程序要对是否为和声状态进行判断，不是对每个音都作rove效果，而是对每组音及和弦作操作。同时要避免连续插入相同的控制器值的现象。

```
(do
 (int int_swch 64)    ;;力度限制值
 (int int_LnR 0)      ;;左右状态记录
(dword dwd_lastTime -1);;判断和声的标记
(int int_is64inserted 0);;中间状态记录
 NIL
 (do
  (getInt int_swch "please input a
switch" 10 120);;或取用户输入
 NIL
  (forEachEvent
   (if (&& (== Event.Kind NOTE) (!=
Event.Time dwd_lastTime))
```

```
    ;;判断是否为音符以及音符与上一音符是否为和声
    (do
      (if (&& (>= Note.Vel int_swch) (== int_is64inserted 0))
    ;;判断力度是否大于限制值，且前面一次操作是否已经给声像置中
       (do
        (insert (- Event.Time 1) Event.Chan CONTROL 10 64)
        (= int_is64inserted 1)
       )
       (do
       ;(pause Note.Vel)
        (if (== int_LnR 0)
         (do
           (insert (- Event.Time 1) Event.Chan CONTROL 10 10)
           (= int_LnR 1);;更改左右状态
           ;(pause "L")
         )
         (do
           (insert (- Event.Time 1) Event.Chan CONTROL 10 117)
          (= int_LnR 0);;更改左右状态
           ;(pause "R")
         )
      );; end if
   (= int_is64inserted 0);;更改中间状态
       );; end do
      );; end first if
     (= dwd_lastTime Event.Time);;记录和声判断值
     )
    );; end the if of NOTE
   );;end for
  );;end do
);; end program
```

cntrThin2.cal

控制器减肥，以前下过一个控制器减肥的程序，但那个是每隔多少个事件就删掉一个，减肥后有些地方控制器变化太厉害了，出现明显的结，于是自己做了一个根据控制器变化的坡度来减肥的程序，用户先输入要减肥的控制器号，然后输入一个坡度值，该值越大两个相邻控制器的变化量就越大。这样可以在兼顾变化的平滑性的同时为控制器减肥，代码如下：

```
(do
 (int int_ctrNum 2);;用户输入的控制器号
 (int int_fade 3);;用户希望的差值
 (int int_lstCtr 0);;上一控制器事件
 (int int_diff 0);;相邻两控制器差值
  (do
   (getInt int_ctrNum "input the Ctr_num" 0 127)
   (getInt int_fade "input a fade value" 1 20)
    (forEachEvent
      (if (&& (== Event.Kind CONTROL) (== Control.Num int_ctrNum))
        (do
         (= int_diff (- Control.Val int_lstCtr));;计算当前控制器与上一控制器的差值
         (if (< int_diff 0) (= int_diff (- 0 int_diff))) ;给负数差值取绝对值
         (if (< int_diff int_fade) (delete));;小于用户希望差值的多余控制器信息就被删除
          (= int_lstCtr Control.Val)
        )
      )
    );; end for
   );; end do
);; end program
```

下面是收集的一些由别人编写CAL程序的介绍，感觉比较有用。基于对原作者的尊重，请不要更改源代码中的任何信息。

C-MPLMIT.cal

作者：Robert

此程序模仿后期处理中压限器的功能对音符力度进行压限，运行时程序首先要求输入一个压缩比，其值为比值的十倍，比如你希望得到2：1的压缩，那么输入20，希望得到2.5:1则输入25；接着程序要求输入一个限制器值，即最大可能的力度值，接着还要输入一个开关，取值为0或1，取1时程序会要求输入压缩的拐点，取0时程序要求输入压缩补偿值。

C-NTROLR.cal

作者：Robert

此程序对控制器值作处理，首先输入控制器号，然后输入处理方法，选0为固定值缩放，要求输入增减的量；选1为范围缩放，要求输入最大值和最小值范围，处理完后控制器会限定在此范围内。

SINGLE.cal

作者：Scott Michael Kemp

截断拈连及叠加音，只能用于旋律声部，当录制时可能音与音之间会有粘连现象，用这个来使音与音之间断开。输入一个间隔的tick值既可。

12TO2.cal

作者：不祥

把大二度范围内制作的弯音值，缩小到适合在八度范围内播放的状态。

CTRLVAL%.cal

作者：Park, Soo Min

这是一个很简单好用的控制器值缩放程序，只要输入控制器号和百分比就ok，要放大就输入大于100的比值，要缩小就输入小于100的比值。

GuitarSplit.cal

作者：Douglas L. Fetter

著名的吉它扫弦程序，非常经典好用，程序要求输入每个音符之间的错开值，单位是tick，然后输入扫弦种类，0=由上向下扫，1=由下向上扫，2=来回扫。此程序有个bug，被处理的音符不能有时值为0的，否则出现变量溢出，出现超长音，解决的办法是先用Edit Interpolate功能将所有时值为0的音符替换成较短的时值，然后再扫。

LEGATO.cal

作者：Joe Pluta

取消拈连现象，使音与音衔接紧凑，使用这个程序要注意几个问题，只对旋律声部使用，如果对柱式和弦使用会导致部份音符时值为0；使用时分乐句来，不要全曲一次处理，否则那些段落衔接处会被音符填满。

NOTE-CUT.cal

作者：不祥

此程序给和弦减肥，输入值为和弦从低到高的音，0表示不减肥，1表示把和弦低音删掉，2表示删掉第二个音，3表示删掉第三个音，4.....最高为6。

Thin Pitch Wheel_x01.cal

作者：不祥

给弯音事件减肥，输入值为每隔多少个事件删掉一个。

点这里进入论坛讨论此文

骆驼加蜥蜴的神秘之音

Cameleon5000简介

卢小旭/文

Cameleon 5000是Camel Audio（骆驼音频）公司推出的一款独具新意的VSTi加法合成器。第一次使用它的时候，一眼就注意到了上面的骆驼和蜥蜴图标，都是特别的动物，看起来有一些神秘感。软件取一个这样的名字一定有它的用意，我们可以想象：骆驼+蜥蜴+5000=神秘+五彩斑斓，呵呵，开个玩笑。

Cameleon 5000使用了比较复杂的合成和创新技术，主要以综合音色、垫底音色以及效果音色为主。预制音色也非常丰富，总共达800多个，此外它还提供了大量效果设置。Cameleon 5000还可以导入WAV和BMP文件，现在它的最新版本为1.6，售价大概在200美元左右。

一、Cameleon 5000概述

1. 主要特点

Cameleon 5000提供了强大并且具有较高柔韧性的加法渐变合成技术。软件的界面设计精美，而且各种功能被分配到了不同的页面，非常直观、给操作带来了方便，让我们可以很轻松地掌握。Cameleon 5000的音色的确是充满一种神秘感，可以调节得特别怪异，也可以听起来亮丽而赋有光晕，这一切都看你怎么去发挥了。下面是它的软件界面：



图01：Cameleon 5000的软件界面

Cameleon 5000可以单独控制每个谐波的振幅、频率和包络。具有多采样合成功能，可以同时存放4个不同的声音程序。有128段的噪音发生器，64段谐音编辑功能。噪音发生器和滤波器都带有“Key Tracking（音调跟踪）”功能。

2. 系统要求

PC：奔腾III 1GHz，128MB内存，Windows98/ME/2000/XP，VSTi或RTAS属主程序。

MAC：G4 733MHz，128MB内存，Mac OS 9/X，VSTi、AU或RTAS属主程序。

二、界面简介以及功能详解

软件界面可以简单分为3个部分：音色模块选择部分、模块调节部分、虚拟键盘部分。如图所示：



图02：Cameleon 5000的各个部分

1. 音色选择区

我们先来看看音色模块选择部分。在下图中，上方用绿色框标出的部分是音色选择区，下方蓝色框标出的一排按钮是模

块选择按钮。如图：



图03：Cameleon 5000的各个部分

上图中，我们给大家标示出了音色选择区的各个按钮和列表。其中如下图所示的下拉列表用于选择音色的类型，单击后会弹出相关种类供我们选择：



图04：音色类型下拉列表

你可以通过此菜单来选择音色的种类。

下图所示为音色选择列表，单击后会弹出具体每个音色的英文：

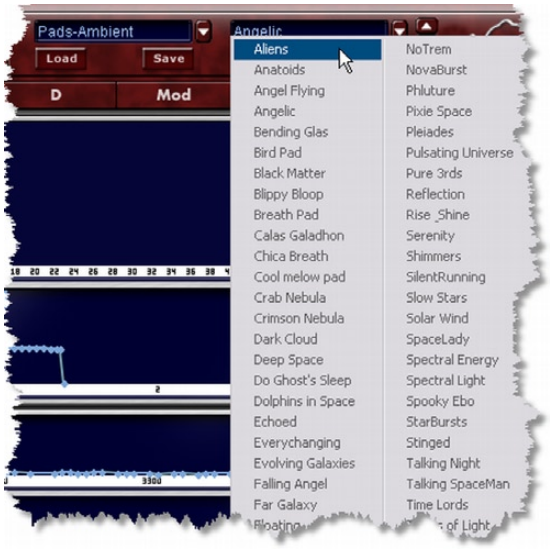


图05：音色选择下拉列表

弹出的菜单中列出的就是具体的音色了，你可以直接在其中选择你想要的。如上图所示为Pads-Ambient种类下的所有音色。

音色选择下拉列表右边有两个按钮，其中上面一个用于选择上一种音色，下面一个用于选择下一种音色。

Load按钮用来读取预制音色，当按下它后会弹出音色读取窗口：

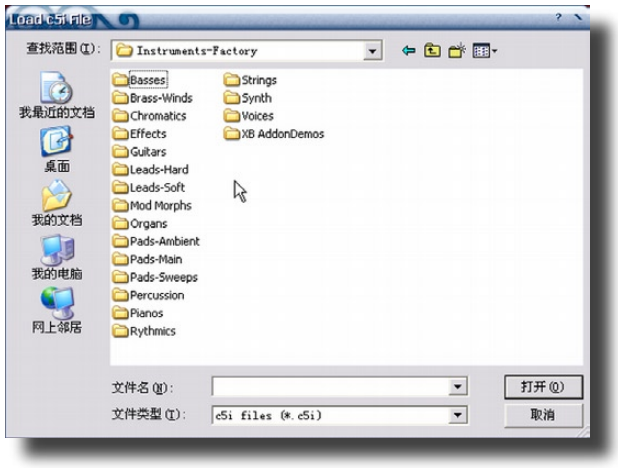


图06：音色读取窗口

你可以直接到音色目录中进行选择。Cameleon 5000合成器的音色扩展名是.c5i，存放的默认目录是Cameleon5000\Cameleon5000Data\Instruments-Factory。可以看到这里的文件夹和前面我们提到的音色类型下拉菜单是相同的，所以通过Load按钮读取音色和用菜单来选择音色是一样的。

Save按钮用来保存自己调制好的音色，按下此按钮后会弹出保存窗口，你可以取名并将其存储下来，下次使用的时候就方便了。

Reset按钮的作用是读取Cameleon 5000安装根目录下的Reset.c5i音色，它是一个为了你自己编辑音色而预制的初始化文件。

Random按钮可以让你得到一个随机生成的新音色，如果你觉得这个音色很好，那么最好是将它保存下来，因为很可能你下次就再也调不出这个音色了。当你按下这个按钮的时候，会看见模块调节部分中的各种设置

参数随机地发生变化。

2. 模块选择按钮与各模块的功能

声音程序模块

从左边开始，A、B、C、D按钮是声音程序模块切换按钮，就如同我们前面所讲的Cameleon 5000可以同时存放4个不同的声音程序，A、B、C、D按钮用于在这4个模块之间切换。如图：



图07：声音程序模块

声音程序就像是在传统合成器中的一个“音色”，换句话说它记录的是声音的设置参数。比如说“风琴”和“梦幻垫底”音色的设置参数就是声音程序。这4个声音程序模块可以说是Cameleon 5000的基础，无论单音音色还是复合音色都是以此为基础构建而成的。

你可以通过模块左上方的Load按钮或Save按钮来读取或保存每个声音模块中的音色，如图：

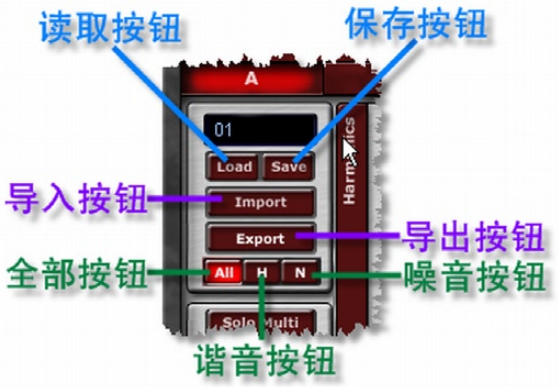


图08：声音程序模块

按下Import按钮后会弹出文件导入窗口：

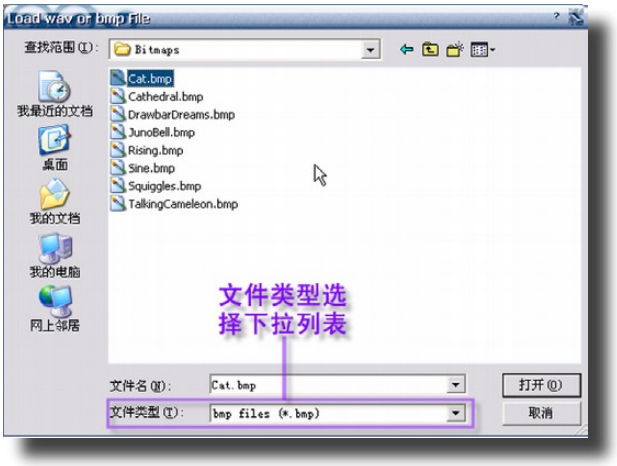


图09：文件导入窗口

你可导入BMP文件（可以是任意的BMP位图文件，甚至是一幅山水画也可以）或者WAV文件，注意其中的文件类型，根据你导入的类型进行选择。文件Import后，Cameleon 5000会对导入的文件进行分析，捕捉文件的特性并将其转化为声音程序。导入WAV文件的时候会稍微有些慢，还会出现一个进度条，注意不要导入太大的文件，如图：



图10：导入进度条

还可通过Export按钮将调制好的声音程序导出成BMP位图文件，这样的文件还能被图片浏览器识别，比如：

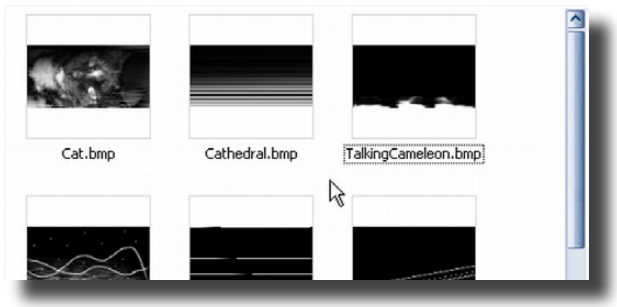


图11：导出的BMP位图

类似的图片在Cameleon 5000的安装目录内也预制了一些，有兴趣你可以试一下。

下边的三个按钮：当按下All（全部）按钮并导入文件时，谐音部分和噪音部分都会被导入；当按下H（谐音）按钮时，只导入谐音；当按下的是N（噪音）按钮时，只导入噪音。

操作软件的时候可能还有个比较难发现的地方就是标尺，它是可以拖动的。上下拖动下图中两个绿框所指的标尺可以缩放它的显示范围，左右拖动则可以将其滑动。如图：

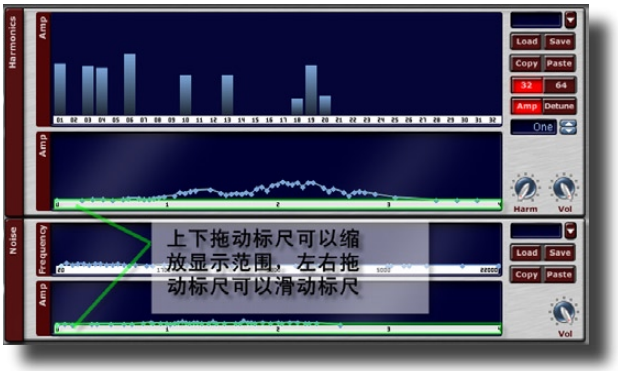


图12：声音程序模块中的标尺

上图中右边谐音和噪音设置区里都有Copy和Paste按钮，它们可以将谐音和噪音的设置参数复制、粘贴到别的声音程序模块中。

模块矩阵

按下Mod按钮会切换到模块矩阵页面，如图：



图13：模块矩阵页面

变化模块

按下Morph按钮可以切换到变化模块，如图：



图14：变化模块

这一模块的主要作用是将A、B、C、D声音程序的声音进行复合和叠加。举个例子，比如说模块A设置的是风琴声音，而模块B设置的是钢琴声音，当Morph Square中如下图所示设置的时候，就会得到一个叠加的音色。

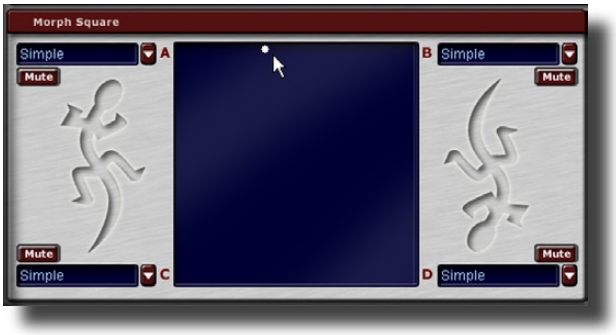


图15：Morph Square中的设置

Cameleon 5000中还有个特别的功能，当点亮变化时间线指示灯后，就可以在下图中有标尺的窗口里通过单击来添加节点（注意此标尺也可以拖动），完成后可以在Morph Square中排列音色顺序，从而合成复合音色。比如下图中设置的音色变化线路可以让音色发生：A音色→B音色→C音色→D音色的变化过程。（“→”表示进行到）。



图16：变化时间线指示灯和音色变化线路

Easy模块

按下Easy按钮可以切换到Easy模块，如图：



图17：Easy模块

效果模块

按下Effects按钮可以切换到效果模块，在这里面你可以给合成的声音或者是预制音色加各种效果。如图：



图18：效果模块

效果模块上一共有6个效果器：Formant filter（共振峰滤波）、Distortion（失真）、Filter（滤波）、Chorus（合唱）、Stereo Delay（立体声延迟）、Reverb（混响）。其中Filter又可以分为HP（高通）、BP（带通）、LP（低通）三种。

假如你选择了一个预制音色，但感觉很干涩，那么可以将Stereo Delay和Reverb效果器打开并进行适当的调节。要打开这两个效果器只需要将它们左上方的指示灯点亮就可以了。

可能你注意到了Morph、Easy、Effects模块中都有下面一幅图：

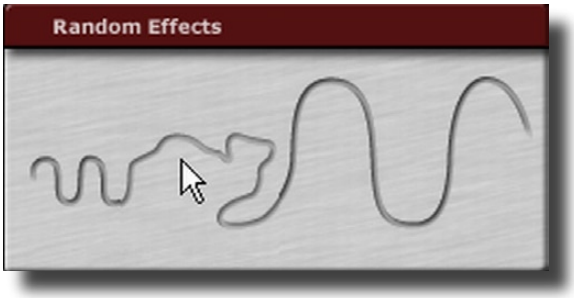


图19：随机按钮

单击图中的骆驼图形，可以让设置参数发生随机变化，它不同于音色模块选择部分的那个随机按钮，它只让所在模块的设置参数发生变化。是不是觉得骆驼音频公司比较逗呢？比较搞笑吧！这个功能我是在第一次使用软件的时候无意间按到骆驼发现的。

三、总结

总的来说Cameleon 5000优点在于操作简单，有很多参数可供调节，预制音色也非常丰富，可以导入WAV和BMP文件，而且还能随机生成音色，可以说音色具有很强的扩展性。无论是制作电子音乐还是做电影音效都是非常不错的选择。相信它能给你的音乐带来乐趣与灵感。F

浅谈流行歌曲的编曲（一）

抒情流行歌曲中的String弦乐组

妹妹/文

流行歌曲的编曲是我们每个人编曲时候的入门功课，编配好的标准则是见仁见智的。我这篇文章里所说的是一些流行歌曲编曲中的“潜规则”，就是大多数编曲者编配流行歌曲的口水歌时所经常采用的方法。……什么？你还不知道？……%￥#&^*@#￥% ……那你就赶快进来看看吧。

这里所说的String不单单指小提琴，它是包括小提、中提、大提和贝提的弦乐组的统称。

一、String在流行歌曲编曲中的常见作用

也就是说，我们经常在什么地方使用弦乐呢？

1.作为长音或和弦，用来增加乐曲声部的厚度

2.作为乐曲的副旋律

3.在高潮时渲染歌曲的情绪

4.在高潮或间奏时作Solo、领奏

二、用软件制作String时必须注意的几点

1.必须画表情线。就是１１号控制器，这是增加弦乐真实感的重要手段。我的所有乐曲只要有弦乐的音轨都是从头画到尾的，画不画表情线在音响效果上绝对不一样！（……什么？你不知道怎么画？……%￥#&^*@#￥% ……那就快点去看看大觉者老师电脑音乐教程第１６课吧）

2.注意各弦乐器之间的融合度。就是小提、中提、大提、贝提的音色之间是否和谐。具体要注意以下几点：

a)避免小2度同时发音。这是最基本的规则，如果你不小心让其中两个乐器一个奏３一个奏４……*&*@^%$*&……那恐怖的音响效果是可想而知的。

b)各乐器声部分明，防止连成一片。各乐器演奏的声部有常规的划分：小提－高音声部；中提－中音声部；大提－中低音声部；贝提－低音声部。在常规情况下你不要让中提奏高音与小提争旋律，也不必让小提奏低音用来加重大提的低音。

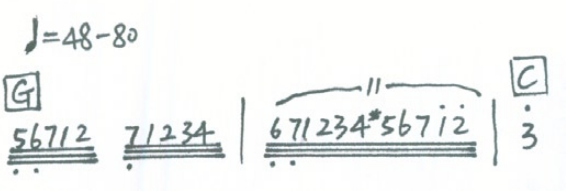
c)注意声部的交替。这条是上一条的延伸。在演奏中要注意各乐器声部的交替。比如一个中提的旋律在它高出了中提琴中声部后，之后的高音旋律就要由小提来完成。

d)注意声部的和声进行。各乐器声部所奏出的音是由乐曲的和声进行决定的。无论你用单独分开的音色制作弦乐还是只用一轨弦乐组来制作弦乐，都是如此。弦乐的各声部要各司其职，在乐曲和声进行的基础上演奏弦乐和弦或旋律。

三、如何用String推高潮

1.用齐奏的方法推高潮。就是弦乐全乐与旋律演奏相同的音。

2.用上行音阶的方法推高潮。（见下方简谱）



例1：手画图

3.在运用上述这种上行音阶的方法推进高潮时，应注意的几点：

a)要知道目的音的音高。在上例中，目的音高就是最后的高音“3”。

b)要根据和声及和弦性质确定音阶中所使用的音。上例音阶的和弦是G，所以需强调G和弦的和弦构成音5、7、2。

c)要根据乐曲风格确定音阶中所使用的音。如果是民乐歌曲，那就要使用五声音阶。

d)音阶中音的密度要根据歌曲的速度定。上例是慢节奏的歌曲所使用的音阶，如果是稍快节奏的，可以使用1 3连音，但最好不要超过1 3连音，因为以本人的经验来看超过1 3连音的弦乐音阶音响效果并不真实。

e)考虑弦乐的乐器法。要注意弦乐乐器所能演奏出的最高音和最低音，不要用键盘弹奏出超出弦乐乐器使用范围的音。

四、String为有人声的歌曲作伴奏的几种常见方法

1.作铺垫（pad string）在软音源插件“超音速”中（Hypersonic），在Pad组有类似的音色，常常在乐曲的低音区奏长音作为乐曲的铺垫。

2.Vc大提琴以琶音音型演奏，增加乐曲的流动性。这种伴奏类似于键盘乐器的左手伴奏。

3.大提琴与中提琴、小提琴的声部交替流动。将上例中的大提琴的琶音音型，分解成大提琴与中提琴、小提琴在低音、中音、高音声部的琶音音型交替演奏。

4.各声部由低到高的声部交替流动。用贝提琴、大提琴、中提琴、小提琴在低音、中音、高音声部用琶音音型交替演奏。

5.小提琴在高声部与贝提琴、大提琴在低音声部奏长音或长线条旋律，中音区的乐器演奏声部交替的流动琶音音型。

五、在Cubase、Nuendo中如何制作弦乐连弓（Legato）

有时我们想模仿弦乐连弓（Legato）的奏法，使我们用MIDI键盘输入的音连续不断，这时就要运用软件中的特殊设置来帮忙了。

在钢琴卷帘谱中选中需要连弓的音——选择MIDI选项——然后依次选择Functions——Legato。

这时我们刚才选中的音就会连续不断的演奏了，并且相同音连奏时声音不会断开。这个选项与量化的区别是：使音与音连接在一起，不受音符时值的限制。

与这个选项相反的还有一个选项，是你发现你演奏的音符音与音都叠在一起了，致使音响含混不清时，你同样选中需要分开的音——选择MIDI选项——然后依次选择Functions——Delete overlaps(mono)，选择这个是单个音，选择它下面的选项Delete overlaps(poly)就是将你选中的全部音都分开。

文章到此你是不是对流行歌曲中String的使用方法有一些基本的认识了呢？在具体的制作过程中我们还是要不厌其烦的追求弦乐音响的真实性，这样才能避免MIDI音乐制作中弦乐音乐不真实的弊端，使我们的音乐作品更加完美。F

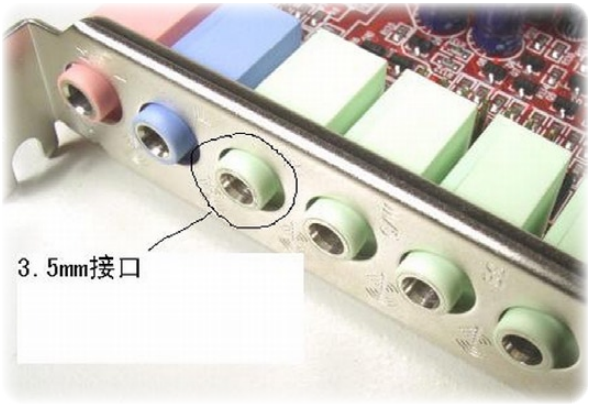
声卡葵花宝典（一）：
声卡接口面面观

王逸驰/文

认识声卡，许多人都是从外观上入手的。如果要说声卡和显卡、网卡等硬件卡相比，外观上最大的区别就在于挡板上的接口了。声卡的接口有非常多的种类，据说任意两种接口的连线共计在100根以上。本文的目的就是要向大家介绍声卡上各种不同的接口，并讨论他们的用途及对音质的影响。

3.5mm立体声接口（小三芯接口）

3.5mm立体声接口又叫做小三芯接口，这是我们目前看到的最主要的声卡接口，绝大部分消费类声卡（包括板载声卡）都在使用这类接口。3.5mm接口提供了立体声输出，因此一般来说支持5.1的声卡（6声道），就需要3个3.5mm立体声接口来接驳模拟音箱（3*2声道=6声道）。而7.1声卡需要4个3.5mm立体声接口（4*2声道=8声道，需要指出的是，Audigy2 ZS等部分7.1声卡采用了特殊接口，因此也是3个3.5mm立体声接口来提供输出——其中一个接口是特制的）。3.5mm立体声接口对于音质没有明显的影响，其主要缺点在于：一、长时间使用后容易接触不良，因此不适合经常需要拔插的使用环境。二、对于高质量音频输出来说，其抗干扰性和立体声分离度无法达到很高的水平，因此在专业级的产品中很少见到3.5mm立体声接口。



RCA接口（同轴、莲花口）

RCA接口是在音响上常见的接口之一，又叫同轴输出，俗称莲花口。一般在外观上分为一红一白两种颜色的接口。每个RCA接口都是单声道的，因此立体声输出需要一红一白两个接口。RCA接口的优势在于可以提供比较好的立体声分离度——2个接口距离相对3.5mm接口比较远，所以受到的干扰相对小。部分高档声卡采用这样的接口。而在HIFI界，一些质量上乘的RCA接口价格惊人。

6.3mm接口（大三芯，1/4）

这是在专业设备中比较常见的一类接口，又叫做大三芯接口。其主要优点在于结构强度高，耐磨损，适合专业场合的反复拔插使用。此外由于内部隔离措施比较好，因此理论上6.3mm接口的抗干扰能力比 3.5mm接口要好一些。6.3mm接口提供了立体声输出或单声道两种输出方式，常见的一般是立体声输出的。比如韩国的waveterminal系列声卡，采用的就是6.3mm立体声输出接口。

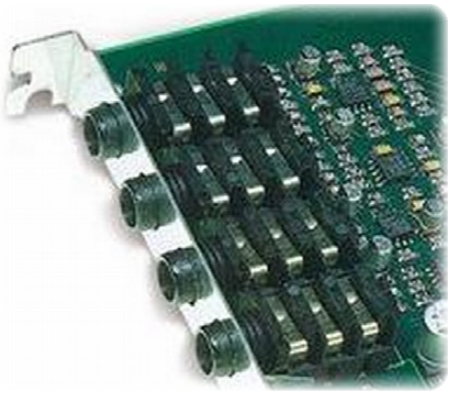
小知识：公头和母头

我们经常会听到谈论公头和母头，另外一种叫法是公插头或母插头。一般来说，能够插入对方的接口称为公头，被插入的则是母头。如果从两性方面来考虑的话，这个叫法非常形象。

一般为了延长接线等等原因，我们会接触到各种接口的公头或母头，有了这点知识，你就不用担忧会搞错了。

1/4 TRS平衡接口

和非平衡的接口一样，1/4 TRS平衡接口提供了平衡输入/输出。TRS的含义是Tip（signal），Ring（signal），Sleeve（ground）。分别代表了该接口的3个接触点。1/4 TRS平衡接口除了具有和6.3mm接口一样的优点——耐磨损外，还具有平衡口拥有的信噪比很高，抗干扰能力极强的特点。对于一个真正的1/4 TRS平衡接口来说，其成本将是非平衡的2倍多。因此采用1/4 TRS平衡接口的设备一般是高档设备，只有在2000元以上的专业卡上可以看到。



MIA声卡上的1/4 TRS平衡输入/输出接口

XLR非平衡/平衡接口（卡农口）

我们在监听音箱或是某些高档音响的背后，都可以看到一个奇怪的3针接口，这就是专业领域中常常提到的XLR接口，又叫做卡农口（实际上卡农口还包括其他的）。XLR接口的接触点可以从二个一直到七个之多，而一般音响器材使用的接头绝大部分都是三个接点。XLR与RCA接口另一个不一样的地方在于：RCA头一定是公头，而座一定是母座，但是XLR接口则不论是头或者是座都有公母之分。相对其它种类接口来说，XLR接口比较紧，接触不良的几率几乎没有。并且由于结构的缘故，在反复拔插的场合磨损很小。并且最关键的优点在于提供了平衡的输入/输出方式。



小知识：平衡与非平衡的传输方式

平衡式传输，指的是平衡传输线里有三芯，一芯负责原始讯号，一芯传输的是反相讯号，最后一芯则无疑就是地线了。一般串进电缆或设备的噪声会出现在+、-两端，对地电压和相位完全一样，这样的噪音被称为共模噪声。由于共模噪声都为同相信号，因此混入原信号中会使得原信号和另外一级信号不能构成反相，因此该信号会被判断为错误。在实际中，由于后端设备只对2芯中完全正相反相的信号产生反应，因此共模噪声无法对后端设备起作用，从而能够很好的抑制共模噪声。因此平衡传输具有非平衡方式不可比拟的抗干扰性能。

平衡式传输是完美音质的一部分，而后端的电路也需要采用平衡式电路设计才能完美配合平衡式传输，两者缺一不可。

一个真正的平衡式电路设计，是将波形的正反相信号分别用一组线路进行放大，因此从成本上考虑，每声道需要2组完全对称的零部件。成本比非平衡方式高出许多。

在电脑音频系统中，一些档次较高的声卡采用了平衡输出。但是需要注意的是，光是平衡输出不能够决定音质好坏，还需要配合拥有平衡输入的音箱才能够发挥优势。因此，片面夸大平衡式传输是不对的。目前一整套平衡式输入/输出的电脑音频系统的价格在5000元以上（声卡+音箱）。

数字传输

上面谈论的是比较普通的一类分类方案。从传输的方式分，音频信号的传输可以分为数字和模拟两类。模拟方式的传输易受干扰，信号随着距离会逐步衰减。而相对模拟，数字信号只传输0或1，因此准确度高。目前声卡上的数字输出口有光纤和同轴两

种，他们都是以光线的开闭来表示0或1的。数字输出将数字信号保真地引出环境相对恶劣的计算机内部，到干扰相对少一些的外部寻找解决方案（如数字音箱等等）。因此目前被广泛使用。



小知识：数字输出的格式 S/PDIF，AES/EBS，MADI

S/PDIF全称为S/PDIF民用接口，英文名为Sony/Philips Digital Interface，很明显它是由Sony和Philips两大巨头制定的标准。就传输方式而言，SPDIF分为输出（SPDIF OUT）和输入（SPDIF IN）两种。就传输载体而言，SPDIF又分为同轴和光纤两种，其实他们可传输的信号是相同的，只不过是载体不同，接口和连线外观也有差异。但光信号传输是今后流行的趋势，其主要优势在于无需考虑接口电平及阻抗问题，接口灵活且抗干扰能力更强。S/PDIF的缺点在于传输距离不远，其信号和后面的专业标准相比易受干扰。

AES/EBS又称为AES3标准，是目前通用的专业标准之一。在专业声卡的数字输出选项中，我们都可以看到AES/EBS。相对于S/PDIF民用接口，AES/EBS的信号强度大，抗干扰能力更强，因此适合更远距离的传输。AES3输出的数据带有自时钟，因此信号稳定程度比S/PDIF民用接口要好不少。另外一个显著特点在于AES3能够胜任长距离的平衡传输。此外AES3的数据报头与S/PDIF民用接口并不完全一样，因此两者虽然有时可以通用，但是由于电器性能的差异，最好避免混用。综合看，AES3减小了通道间的极性偏移、通道间的不平衡、噪音、高频衰减和增益漂移等问题造成的影响。

MADI即AES10标准，是AES3标准的延伸，全称为多通道数字音频接口（Multichannel Audio Digital Interface）。它可以在50米的距离内通过一根带有BNC端口的电缆串行传输56个通道的线性量化（PCM）音频数据。MADI是异步工作的方式，也就是音频信号的时钟不是和数据在一起被传输的。 MADI接收机要从传输来的数据中抽取时钟信息，以便接收机的能够得到正确的时钟信息（时钟同步）。为了保证这一点，MADI协议固定每帧至少传送一次10比特的同步标志。

外部连接口

USB接口

我们经常看到一些外置声卡采用了USB接口来连接到计算机。采用USB接口有两个好处：一是即插即用，设备能够被自动识别。二是方便快捷。因此不少外置声卡以此作为最大的卖点。不少人认为外置就是音质好，其实不然，至少对于USB接口来说，音质并不一定好。首先，和PCI

总线相比，USB接口的传输速度有限，这样将导致处理繁重的数据资料时候会有延迟，延迟对于音频来说是相当致命的。其次，USB总线并不是一个很稳定的总线，传输速度不稳定，并且由于在系统中的优先级一般，易受系统的影响。因此，对于音频来说，USB总线并不太适合，如果要外置，不如IEE1394接口。

IEE1394接口（火线接口）

IEE1394接口是一些专业音频卡常用的接口，就是人们常说的火线接口。相对于USB，IEE1394接口具有传输率高，传输稳定等显著的优点。目前各家声卡厂家都在大力推广采用火线接口的声卡。因此今年很可能是“火线”年。

外置盒接口

许多声卡都采用内置PCI卡+外置盒的结构，比如著名的Sound Blaster Audigy2 ZS系列。这里说的接口就是指连接内置PCI卡和外置盒的接口。有人会说了，这个接口有什么重要的作用啊？其实这个接口基本决定了声卡的音质。外置盒接口具有模拟和数字两种形式。如果是模拟的，则由于传输距离长，易受机箱内部干扰等因素，音质会反而比原先内置的音质差。如果是数字的，则音质由外置盒来决定，这是最好的情况了。因此，看外置盒的接口，可以粗略判断音质，如果看到是模拟接口，基本就可以断定外置盒是个摆设了。



Audigy2的外置盒连接口
采用数字传输，受到的干扰小。



Waveterminal 192M的外置盒连接
由于采用的是模拟传输，因此音质受到影响。

MIDI相关接口

MIDI/游戏口

这是目前我们常常看到的接口，主要用于接驳游戏控制器和MIDI键盘等设备。



MIDI接口

这是目前常用的MIDI接口，在专业设备上都可以看见，采用的是5针圆形接口。

门子缩混系列连载：

在笼子里跳舞（二）

门子/文



我们需要什么样的分轨

又到了一月一次的催稿时间，老大的压寨夫人对我总是说老大催稿的事情表示强烈不满，没办法啊文章开头总要讲点什么来引出下文嘛，文笔太差只好说催稿喽～另外我女朋友对我总是好像不关心她也表示强烈不满……9月18号是她的生日，在这里提前祝她生日快乐，门子不讲爱情故事，不过门子很爱她呵呵。

注：本来想单独写一篇拾音技巧，但是由于工作关系，除了录人声和吉他比较多以外，很少亲自录别的乐器，所以写出来也无非是和现在到处都能找到的大小ab，xy啊这种录音技巧文章一样。和别的文章讲的差不多——这正是门子“在笼子里跳舞”这套教程里希望能避免的。所以，在此就不在重点讲缩混的文章里面讲拾音了，只是从出发点上提一提。相信会有人写出更棒的拾音技巧文章的。

瞎扯半天，言归正传。上回书说到到底什么是缩混，从比较基础的方面开始谈到缩混。那么今天，我们就要以上次说的为前提，继续来写点东西。这一篇的内容是“我们需要什么样的分轨”。把这个作为缩混教程的其中一部分会不会篇幅过多呢？呵呵，俗话说巧汉子做不了无米之炊，分轨质量在缩混时的地位是非常重要的。无论是音源导出的分轨，还是录音得到的分轨，如果质量实在太差，大多录音师是不想收的——我大不了少作一份买卖，可是把一个很烂的分轨做的很好，这是绝对费力不讨好的事情。观众才不知道到底是录音师手艺不佳还是分轨质量出了问题。我接过这种活，最后缩完只拿钱，不写我名字……

分轨的得到无非两种：midi音源的导出与录音直接得到。但无论是如何

得到的这个分轨的，都要满足如下的两个条件：

1.尽量要作出后期做不出，或不方便做出的效果。

2.尽量不要加入妨碍到录音师摆放此分轨的自由的效果。

为什么会有这么两个条件呢？我们不妨先说说第一个条件：

nisoxboy：我知道！前期什么效果都不加，全都放在后期加！

这可不一定，某些效果在前，后期的得到是不同的。什么是后期不方便做出来的效果呢？举个最简单的例子，一首歌的录音分轨里面没有人声，那你是再厉害也缩不出人声的……

大睡者：我晕！门子你这完全是废话。

呵呵，这么说好象是废话，可是很多情况下，人们却毫不重视这句话，咱们再举个例子：这是简单的歌曲，其中有一个电bass。（点这里下载MP3演示文件）

无论你对这个歌曲进行什么样的缩混，你都会发现这个bass就好象一个中声部一

样，一点也没有起到它应该起到的作用。这是为什么呢？通过频谱分析就能发现，这个bass缺少应有的低频。（100hz以下的东西很不扎实。）你就是用eq把低频拉10db出来，也没用。

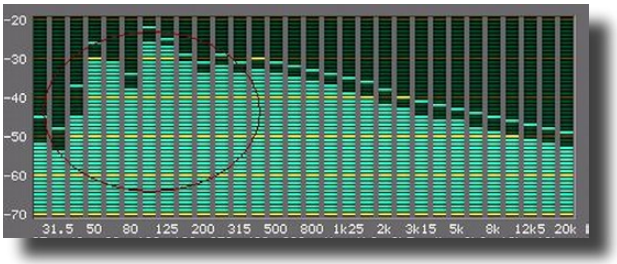


图1 频谱看不出来，但是这里的声音很虚，提不出来



图2 提成这样也没啥戏

大睡者：哇！！原来它低频的声音太少了，怪不得这个歌怎么缩也沉不下去呢！！那有什么补救方法么？

其实，面对这样的声部也不是没有补救方法，只是就算进行了补救，效果也不会达到最佳效果。最好的方法是——重新让编曲换音色。这就是所谓“作出后期作不出来的效果了”。其实道理非常简单——“后期”对声音的处理是有限的，某些声音只能在前期得到，所以我们就要尽量在前期得到。某些东西，比如你人声唱的不好希望在后期得到补充，这是很正常的想法。但是处理也是有限度的，目前的科技水平还作不到把门子的破嗓子修成帕瓦罗蒂……所以说，我们在前期，也就是分轨制作的时候，绝对不允许抱有“前期随便做，后期修就可以了”的想法。很多初学者都把自己制作，录音上的瑕疵归到“后期”的处理上去，仿佛什么声音一走“后期”就变的怎么怎么厉害了。这个风气很不好，“后期”的目的其实最终还是咱们上回书说的“把声音合理地放在笼子里”，而并不是加些处理就行了的。
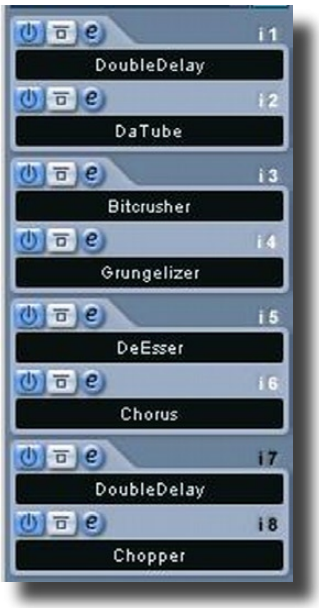


图3 分轨如果实在太差了，任你加什么效果器都没戏

那么，到底什么样子的处理，后期不方便做到呢？某些情况是明摆着缩混的时候做不出来的，比如唱错词啊，midi做的不像等等。但是上升到一定高度，就会发现所谓“缩混时做不出来”实际上不存在一个明显的边界——某些后期效果可能这个效果器能做到，那个做不到，某些效果这个人做的到那个人做不到。某些效果以前做不到现在做到了。比如刚才咱们说到的bass低频缺少重要成分这个问题。如果有可靠的低频激励，

也是可以激励出一些低频成分的。

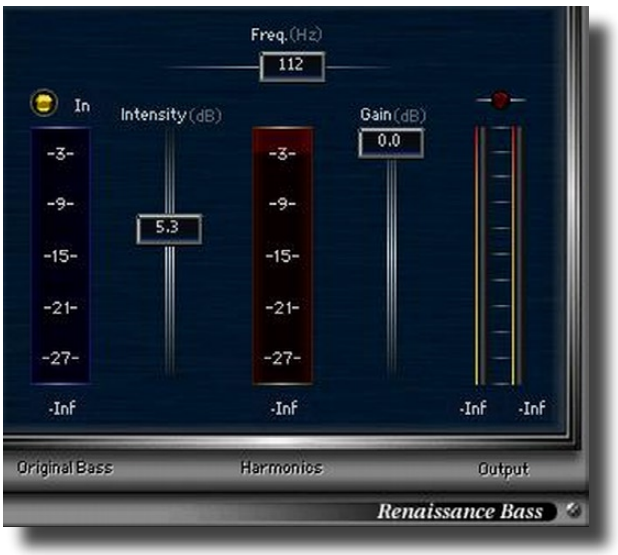


图4 waves这个低频增强有时候还是能起到一定作用的

再比如很久以前，人声跑调是修不了的，人们只能以“不跑调”当作人声前期的重要原则。现在随着拉跑调机的发明，人声录音也是完全允许跑调的了。



图5 大家熟悉的Auto-tune

那么，如何来划分什么效果前期做什么后期做呢？我们需要明确的是：本该在前期分轨时候解决的问题，越拖到后期，效果越不如前面。比如吉他录音的时候使用一把很普通的吉他，我们也是可以在缩混的时候把它变为一个好听的声音的。但是这个效果比你直接使用一个好吉他录音要差很多。（点这里下载MP3演示文件）

如果把分轨的得到划分为音源导出和录音的话，我们不妨先看看真实录音得到的声音的好的分轨标准：

nosixboy：标准我知道我知道！不就是不加任何效果器，用mic录制最真实的声音。

呵呵，何谓“不加任何效果器”呢？我们要知道的是，mic就是个最厉害的效果器。它永远都在美化声音。假如它真的只是在还原声音的话，那录音棚还需要那么多mic作什么，直接买一根最最最还原声音的mic不就行了？

大睡者：mic是效果器？难道mic还能当eq使用不成？

呵呵，没错，假如我想得到一个感觉亲切的人声，你怎么做？

大睡者：额……用eq拉高频啊！

呵呵，如果是我的话，我会首先想到让歌手站的近一些。

nisoxboy：这……两者效果不一样吗？

这个……门子也说不清为什么它不一样，但是反正它的效果就是用mic做的比缩混时候用eq做的好。说mic是个eq其实并不贴切，但是它确实能做到很多频响上的改变，这种改变是eq很难实现的。mic摆位和选择是个绝对绝对的艺术，这里只能简单来讲。首先对于单个mic来说，距离发声源多远，是否“偏轴”，以及mic的种类，是3个影响效果的重点：

1.距离发声源距离多少：

离发声体越近，细节感越强，整体感越差。

离发声体越远，整体感越强，细节感越差。

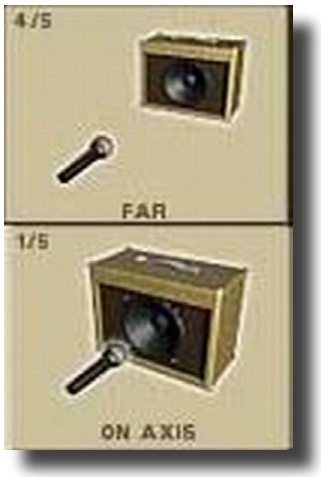


图6：距离位置关系

大家可以听这两个录音，一个是mic摆位距离比较近，一个距离较远，第一个感觉完全好象是一张嘴在你面前说话，另一个就是一个人在你面前说话。（注意！摆位远近的标准是相对的，尤其在吸音不佳的环境下，离远了首先带来的不是整体感而是环境混响）。这就是所谓整体与细节的把握。有点鱼和熊掌的感觉。具体要用哪种声音，要看你的歌曲要求了。但是无论怎样，这种感觉依靠eq是很难调出来的。（注意，mic离嘴距离的远近与缩混时人声的远近无关！）(点这里下载MP3演示文件)

2.偏轴：

正对声源叫做（on axis）斜对叫做（off axis）。



图7 就像这种

它们的区别大约在于：正对的声音很冲，炸。斜对的声音较为平滑。

这在对于吉他录音，尤其是电吉他录音很有讲究，经常将这两种方法结合使用。(点这里下载MP3演示文件)

这种效果，用eq是很难实现的。听起来好象只是高频多少的问题，可是真正尝试一下就会发现，不是那么回事。（在电吉他录音里有这样一个说法：能用mic摆出来的，就不用eq。）



还有一个概念就是mic的种类，不同的mic会让声音发生不同的变化，比如经典的sm57，为什么会是录制电吉他的利器呢？因为本来它的高频响应很一般，但在使用大声压来“激”

它的时候，它的高频会产生一种很“炸”的感觉。非常适合电吉他的特点。

这只是一个例子，在拾音中，还有很多用mic做出来的效果（人声相对较少，主要是乐器），比如利用mic的近讲效应增加低频等等。在此就不一一列举了。

(点这里下载MP3演示文件)

话放什么的设备，与mic同理，实际上好的话放也是某种意义上的效果器，把gain旋钮拧到不同的地方，声音就会不同。

所以说，一些前期和后期都能做出的效果，从前期的角度去做效果会更加好。常见如下：

1．录音时候保证把频段录饱满了，不要出现bass缺低频之类的现象。

2．动态不宜太小，适当的进行一些压缩是很合理的。

3．某些特殊的声音细节。

（点这里下载MP3演示文件）

再回过头来看看音源导出的声音，其实对于采样音源来讲，它们无非就是一些采集好的声音罢了，也很符合我们前面拾音的特点。现在的音源，尤其是大一些的软件，现在的音源，尤其是大一些的软件，基本都是频响甚满动态给足的。所以基本用不着再进行什么处理，只要把midi做好了就行了。另外很多人存在“音源到底导出单声道还是立体声”的问题，其实那要看音源，比如它本来就是个单声道的鼓，那就导出单声道就行了，本来是立体声的bass，那就导出立体声。至于最终我们缩混的时候使用单还是双，那是后话了。如果音源是单声道的，后面我们有办法把它变成立体声的，但是如果音源本来就是立体声的却导出了单声道，那后期再处理成立体声的效果就不如原来好了。（点这里下载MP3演示文件）

如果实在不确定自己的音源是单还是双，可以借助相位仪工具，waves里面就有。

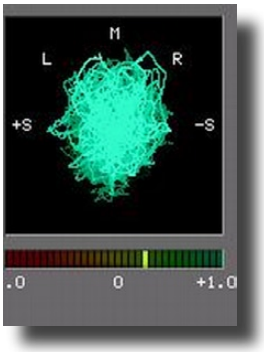


图8 这是我的宝贝9632的相位仪

大睡者：我知道啦！！！既然前期做出的效果比后期好，那我就所有效果都在前期加，那后期就用不着加效果啦！！直接在前期合起来就行了！

Pia pia！！！你没看见还有第二个条件么？——尽量不能妨碍到混音师摆放此分轨的自由。

什么样的效果就妨碍了混音师的自由了呢？最简单的例子就是只给混音师一个已经混好的东西……那他是无论如何都不可能再进行缩混了。进行母带还差不多。我们应该清楚为什么缩混需要把整个歌曲拆分为每个独立乐器的分轨。

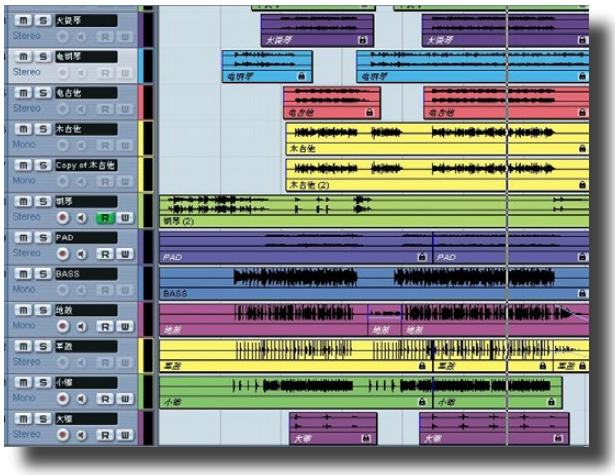


图9 分轨图片

大睡者：我知道！是为了单独加效果器！

呵呵，上回书我们就已经说过了，不加效果器也可以缩混，而且现在几乎所有音源都可以直接在编曲状态下加效果器，用不着单独导出加效果器啊。单独加效果器只是我们分轨导出的其中一个目的。

那么，到底是为什么单轨导出呢？其实，“分”轨的最大目的在于：可以让

录音师真正自由得在频率和响度构成的笼子里面进行摆放。还记得上回我们说过的么？舞蹈家可以随意编排舞蹈演员的动作，而现在我们只能让演员们进入一个笼子跳舞。如果已经完全规定好了每个演员的动作，不允许舞台编剧再次修改，我们就没有办法让他们在笼子里跳舞了。编曲也是同理，如果已经把所有声部混在了一起，录音师就不能在音频的笼子里面进行摆位了。（这里的摆位是指在eq上，音量上，和pan上的综合摆位，而不仅仅指pan上的。）

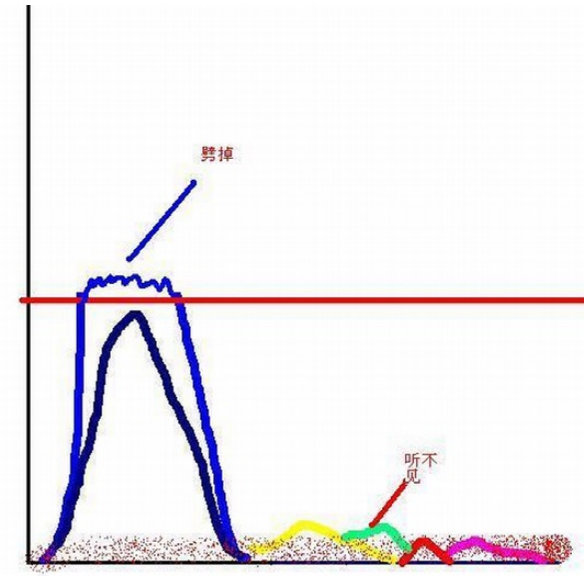


图10 上期说过的“笼子”

nosixboy：可是上回我没有把鼓组分开给混音师，缩出来也不错啊。

那是因为编曲人的摆位想法和混音师基本一致， 所以无须再次进行摆位了。实际上比较好的编曲人已经在编曲的时候考虑到了缩混的要求。而且这也是一个效率与质量的感觉。如果编曲人觉得某些声部有把握不用拆分，那不拆也可以。比如某些鼓组loop，仅仅是用来打底，让音乐带有一些稳定前进感的，就可以不拆。但是大多数分轨还是要掰开的。

除了轨道要掰开，分轨的pan需要录音师来调整，所以分轨是不允许带有pan信息的，也就是导出的时候要把分轨的声向调到中间。在我们上一篇文章里面的笼子理论已经提到，还有一些效果会干扰到缩混的摆放。最典型的就是混响，分轨不加混响可以说有点成了规矩了呵呵，其实这个的主要目的就是不干扰录音师的摆位。此外，EQ的高低切之类的手段也尽量不要使用，会影响缩混时候的频率调节。音量也要按分轨打到最大，交给缩混者来进行音量的平衡。

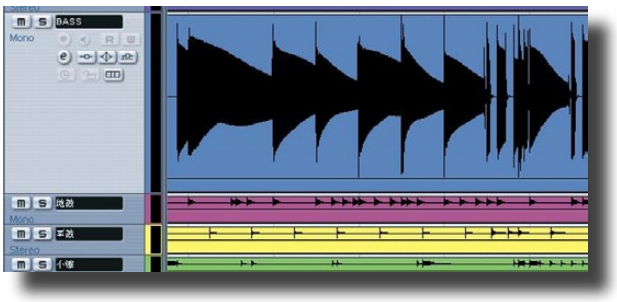


图11 这种分轨看着就爽！

最后总结一下，所以说我们导出分轨不用把所有的效果都关掉，尤其是对于某些音源来说，关掉了反倒不好了。并不是所有的分轨都不加效果。而且不能加入任何干扰录音师进行缩混的效果。今天就先说这么多吧，谢谢诸位捧场！下期只要老大也不要忘了催稿子哦！

专业音乐软件开发入门（五）

解读XSynth（上）

Seer Xillion/文

从本期开始，正式开始讲述VSTi的编写。之前对于VST编写有任何问题，都可以跟我联系，可以在MIDIFAN上讨论，也可以写信给我。

跟VST不一样，VSTi除了需要学习接口知识以外，还需要额外学习一些写合成器“必备”的知识，而且有些内容还跟MIDI信息（事件）相关，故限于篇幅，本期只能讲解VST2.4的SDK自带的一个VSTi了。

不要小看本次的讲解，它将成为今后写任何VSTi所必备的知识。不然到时写起来会觉得“无从下手”！！

首先打开如下工程（我将假设您是把VST SDK 2.4放在“老地方”的）：

D:\TEMP\vst2.4\public.sdk\samples\vst2.x\win.vc6\vstxsynth.dsp

打开工程后，为了方便，先按照先前介绍过的方法让其能自动在编译完成后复制到VST插件所在的目录下（不记得的话找前几期的来看……）

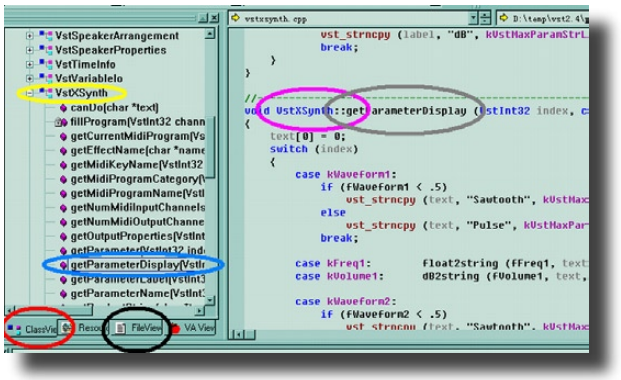
作为一个预览，首先将其编译出来，并在Nuendo下看一下。

我在此先做个简要的说明：

该VSTi是个单复音（Monophonic）的合成器，即所谓的复音数为1,很可怜吧，但没有关系，作为一个对VSTi认识的通道，这已经足够了。从界面上看，上方的三个控制条控制的是左声道，下面的是右声道。每个声道可以分别控制自身的波型、频率偏移以及音量这三个参数。波型方面可以有两个选择——锯齿波和脉冲波。

对插件的基本功能有所了解后，再来看代码就好解释了。

本次讲解还将介绍一种定位代码的方法，该法对于程序比较复杂的情况，如果按照前几期那样讲可能讲到天亮了也许大家也找不到据说的代码段。



现在比方我说要找如下函数（即上图右边显示出的函数）：

首先按Alt+0（字母P左上方的数字0）让VC++显示上图左边的视图（如果没有显示的话），然后注意到红色圈的那一页“ClassView”，点击它打开的是“类视图”，一个C++程序经常是由很多类组成的，比如本例中我们要找的函数所属的类：VstXSynth （紫色圈）

图中左边的各个类的先后是按照音序排的，所以很好找，而点击想到的类的左边的加号（上图已经展开，就是减号了）就可以展开该类的所有函数和变量等等。

然后是找到想找的函数名，本例就是图中右边的函数名：getParameterDisplay（灰色圈）

由于展开后的名字也是按照音序排的，所以也很容易找到要找的函数，双击其即可（图中蓝色圈）。

简单地说就是把“：：”左边的作为

“类名”，把“：：”后面的作为函数名来找。

现在大家自己试着用如上方法找到我们要找的第一函数（找不到的回幼儿园重修及格后再回来看！！）：

```
VstXSynth::VstXSynth (audioMasterCallback audioMaster)
: AudioEffectX (audioMaster, kNumPrograms, kNumParams)
{
……
  if (audioMaster)
  {setNumInputs (0);  // no inputs
   setNumOutputs (kNumOutputs); // 2 outputs, 1 for each oscillator
   canProcessReplacing ();
   isSynth ();
   setUniqueID ('VxS2');// <<<! *must* change this!!!!
  }
  initProcess ();
  suspend ();
}
```

上面省略号部分大家可以根据以前对VST的学习自己去掌握，这里只讲解跟VSTi有关的内容。

首先该函数大家不知道还记不记得，叫做“构造函数”，即当VstXSynth这个类“出生”时会被调用的函数，也就是当你插入一个VSTi到Nuendo中时这里就会运行了！

在 if (audioMaster) 中的那些语句的主要作用是设置该VSTi的基本属性，我们来看看以前学VST时没有说过的：

```
canProcessReplacing ();
```

该句直译就是“能做替换式处理”，说明这个VSTi实现了“processReplacing”这个函数，大家应该对该函数还有印象吧！

相对地，如果这句是如下的样子，情况就不一样了：

```
canDoubleReplacing ();
```

以上的意思直译是“能做双份的替换”，很难理解吧，如果去看SDK的帮助文件，就能知道它真正的意义是什么了：“能做双精度的替换式处理”。

所谓“双精度”，指的是比相对的“单精度”具有更高精确度的运算。也就是大家近来常听说的64位精度！而我们熟识的“canProcessReplacing”就是单精度的，不过只有2.4版的SDK才能支持64位精度的处理。

```
isSynth ();
```

这一句的出现，表示该插件“是合成器”，即它是VSTi。当Nuendo发现这个插件是VSTi时，则可以令这个插件出现在合成器架子中，让用户插入作为Instrument用。

```
setUniqueID ('VxS2');
```

这一句表示该插件的“唯一标识”叫什么名字，通常当你开发一个自己的插件要拿去给人家用时（不管是VST还是VSTi），都一定要更改这个名字，否则全世界都在开发功能不同的“VxS2”插件了。

比如假如我们上回开发的CrossDelay，要写成“标识”的话，可以考虑写成如下这个样子：

```
setUniqueID ('XDly');
```

因为习惯上，Cross会被表示成“X”，而Delay的标准缩写是“Dly”。不要写连自己都不知道是什么东西的代码！如果我们下一期要写一个“红白游戏机的VSTi”，则可以写成如下这个样子：

```
setUniqueID ('NESS');
```

其中“NES”就是表示“任天堂娱乐系统”，而“S”是“Synth”的缩写，大家还记得“YAMAHA S-XG50”这个名字吧？其中的“S”就是“合成器”、“软音源”的意思。

当然如果你认为用其他表示法也能表明是这个意思，也可以写成那个样子。

接下来看到后面这行：

```
initProcess ();
```

该函数从名字上看就是“初始化处理”，也就是在正式开始处理之前做一些“准备工作”。以下是这个函数的内容，大家请务必找到：

```
void VstXSynth::initProcess ()
{
  fPhase1 = fPhase2 = 0.f;
  fScaler = (float)((double)kWaveSize / 44100.);
  noteIsOn = false;
  currentDelta = currentNote = currentDelta = 0;
```

上面这一部分的内容简单来说就是把计算时作为状态用的变量设置成默认值（初始值），其具体含义在用到时再详细解说。

从这里开始，我们将讲述更多关于C/C++语言的知识，因为有些知识不讲的话还是很难写程序的。我们还是先慢后快吧！

下面这一块，是预先做好两个“波表”，一个是“锯齿波”，另一个是“脉冲波”。

```
VstInt32 i;
```

这里是定义了一个变量，叫做“i”，其类型是VstInt32，直译就是VST的32位整数，大家认为它是个整数就好了。以后看到定义变量我就不解释了，定义一个变量是迟早要使用的，所以我会直接在使用到这个变量的时候来说。比如后面的“wh”这个变量。

```
// make waveforms
```

这种用双左斜杠“//”开始的内容，在C/C++中叫做“注释”，所谓注释，即其内容是为了方便阅读程序的人来看懂代码，而电脑不认的。比如你写成下面这个样子，VC++也只会“Bypass”它而已：

```
// 我是Steinberg公司的一员！！
```

所以本处的注释能很好地帮助我们读懂以下的代码，直译就是“制作波型（们）”，哈哈！

```
VstInt32 wh = kWaveSize / 4; // 1:3 pulse
```

这里又定义了一个变量，其意义就是表示1:3的脉冲的宽度，用到时大家就明白了。

这里跟先前介绍过的while一样，也是一种循环的表示方法。在C/C++中表示循环的方法一共有三种，从某种意义上说，用任何一种循环都可以代替另外两种，只是一定会有一种使得设计起来更直观易懂。

这里出现的循环叫“For循环”（以前介绍的叫“While循环”），很无聊吧，中国的教学就是爱把一些本来很简单的东西“术语化”，让人感觉很专业！！还有一个叫“Do循环”的！

以下这个循环表示i从0开始，在小于kWaveSize的情况下做花括号中的内容，做完后做“i++”这件事，然后再看是否还满足“i < kWaveSize”，是的话再做一次花括号中的事，然后再“i++”……

可以看出，当i在随着做“＋＋”这件事的时候，就会每次增加1，终有一天会变成跟kWaveSize一样的值，这时这个循环就算搞定了！

所以当看到下面这种形式的句子时，基本上可以认为就是“花括号中的事做kWaveSize次”。至于kWaveSize是多少，大家看到小写k开头的东西不觉得很熟悉吗？用前几期介绍的方法找到它的值吧！！（前面上课睡觉或者课后不举一反三的话现在就亏了！！提示一下，用Find In Files或者在kWaveSize上点击右键选择“Go To Definition of ……”）

```
for (i = 0; i < kWaveSize; i++)
{
```

花括号中一共做了两件事，先看第一件——做锯齿波的波型：

sawtooth[i] = (float)(-1. + (2. * ((double)i / (double)kWaveSize)));

为什么这样就能做个锯齿波呢？先从这一句上看看它做了什么事！

这句虽然看上去复杂，但把它“数学表示化”后就很看清楚了：

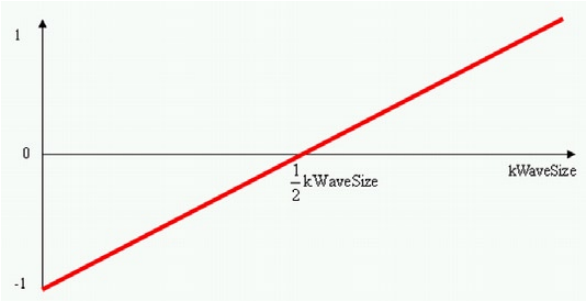
$$\text{sawtooth}_i = -1 + 2 \bullet \frac{i}{\text{kWaveSize}} = \frac{2}{\text{kWaveSize}} \bullet i - 1$$

是不是跟中学时的什么很像哩：

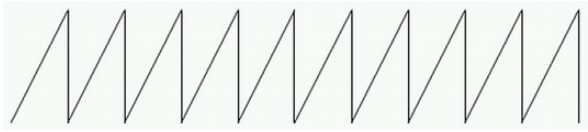
$$y = kx + c$$

没错，这是一条直线的方程式，不过在本例中，由于i是有范围的，所以它是一个线段！

其图像如下所示：



虽然咋一看不像锯齿，但仔细看看我们认为是锯齿的图就不这样觉得了：



怎样，是锯齿波吧！！只要我们生成了一个周期的波型，剩下的就可以通过将其复制或者高级复制来生成了。

接下来看看脉冲波的情况。

pulse[i] = (i < wh) ? -1.f : 1.f;

嘿嘿，这里出现了一个奇怪的语法，待我讲来……

C ? A : B

这在C / C++中叫做“问号”运算符，所谓运算符，就是像＋－×÷那样的运算用的符号。那它的运算法则是什么呢？

它一共有三个元素：C、A和B

其中C是条件（Condition），A和B分别是两个值。

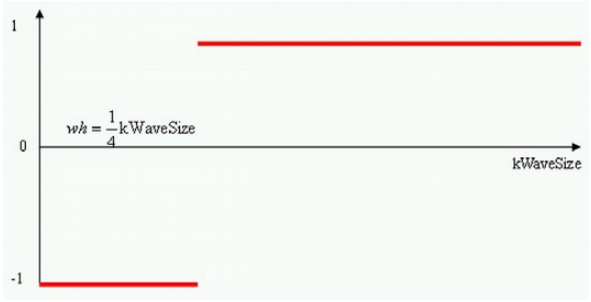
如果C为真（True），即不等于0的时候，或者说判断条件成立的时候，这个运算的结果是A，否则就是B。

比如下句的情况中C就是(i < wh)，A就是-1.f，B是1.f。

直译就是当i小于wh时pulse[i]为－1，否则pulse[i]为1。

这又是怎样的景象呢？

我们刚才知道，wh是1 / 4的kWaveSize，套用刚才的图像可以知道，在1 / 4图像的地方，图线是－1的值，也就是在底端，而从1 / 4那里开始的所有右边的值都为1，即最顶端。怎样，一个1 / 4宽的脉冲波型图是否已经展现在您的大脑中了。虽然我很不喜欢，但还是画一下吧：



是不是又觉得跟我们熟识的脉冲波不像？嗨，再画一下吧：



}

大家是否觉得数学还是有点用的了哩？

所以如果大家要写一个其他形状的波表，只要灵活运用数学知识就可以了。比如要做个正弦的波表，看看推导过程：

1.首先是公式

$$y = \sin(x)$$

2. 再看 x 的取值范围：$0 \sim 2\pi$

3. 再看我们i的范围：0～kWaveSize。也就是说，当i从0变到kWaveSize的过程，

就是相当于sin函数的 x 从0变到 2π 的过程。因此，很容易得出如下结论：

$$x = \frac{i}{kWaveSize} \bullet 2\pi$$

整合起来就是如下的样子：

$$y = \sin(\frac{2\pi \bullet i}{kWaveSize})$$

根据之前的学习，可以很容易写出想要的程序：

```
sine[i] = (float)(sin(2 * 3.1415926 * i / kWaveSize));
```

为了验证，大家可以暂且把sine[i]写成pulse[i]，然后编译，会发现如下一个错误：

```
D:\temp\vst2.4\public.sdk\samples\vst2.x\vstxsynth\source\vstxsynthproc.cpp(66) : error C2065: 'sin' : undeclared identifier
```

直译就是“sin是没有定义的标识”，简单地说就是VC++不认识sin了。

其实C/C++是能算sin的，只是由于它并非在一般情况下常用，就需要额外告诉VC++：“我这个程序要做些科学计算”，告诉的方法如下：

首先拉到当前编辑的文件的最上方，是不是能看到下面这行啊：

```
#include "vstxsynth.h"
```

好的，在它的下面增加如下这行：

```
#include <math.h>
```

这时再编译就通过了！还不快点去Nuendo中听听是不是正弦的声音！

今后，如果你的程序要进行这种运算，就记得加上这一行哦……

再回到刚才的程序，细心的朋友也许发现了，为什么前面的代码中在计算锯齿的波的时候那么多double，而在计算脉冲的时候就不用写double了呢？因为在C/C++中，整数和浮点数不仅仅是我们理解的那样一个没有小数点一个有小数点那么简单的。为了让大家在编写时尽量不出错，在这里对一些关键点简要说明一下：

1.对于除法运算来说，整数除以整数的结果是整数，而只要有一个不是整数，其结果就是浮点数，这是按“大”对齐的。所谓大，就是浮点比整数大，双精度比单精度大……。看下面的例子：

a)int a=12;

int b=5;

那么a / b的结果是2。

b)double a=12; (float和double都表示是浮点数，前者是单精度的，后者是双精度的)

int b=5;

那么a / b的结果是2.4。因为a和b有一个不是整数。

对于a)的情况，如果写成 a / (double) b 的话，其结果就是跟b)是一样的，因为那个加了括号的double，表示强制类型转换。也就是强制把b转成double型，构成除法运算中，其中一个为浮点型。如果要强制转换成其他类型，也可以这样用。大家看到的(float)就是把后面的双精度浮点数转成单精度浮点数。

这就说明了为什么计算锯齿时要用double了，其实如果它不加那么多括号的话也是完全不用加的，因为先跟2.（注意2后面有个小数点，是浮点数！）相乘后的i也已经满足“a和b有一个为浮点数”的条件了，因为乘也是一样的，两数相乘，有一个是浮点数则整个结果就是浮点数。而乘法和除法由于是同级运算，所以会先算，这个跟数学是一样的。所以在我们写的正弦波的例子的时候，就把那些累赘的double和括号都

去掉了。

2.最好不要比较两个浮点数是否相等。比如a>=b, a<=b, a==b分别是表示对a和b的大小比较，即大于等于、小于等于和等于。因为在计算机中，浮点数是不精确的！原因就不多说了，总之要比较一个浮点数是否等于一个数，保险且推荐的做法是：

```
( a - 10 < 0.0000001 ) && ( 10 - a < 0.0000001 )
```

其中，“&&”表示条件与，也就是参数判断的两个条件要同时成立才算成立，即所谓的“与”。可以看出，上述的条件能判断出a是否很接近10。所谓很接近，就是跟10的差很小，在本例中，认为0.0000001是很小的数。

这就是为什么它不把i和kWaveSize定义成浮点数而是定义成整数了，因为整数的比较是绝对的，而且速度上也比浮点数要快。

所以，在写代码时，能用整数的尽量用整数！不然等你的VST或者VSTi功能很强大时，你会发现你的声音是断断的……

所以，如果你想要设计一个使用“波表算法”为基础的VSTi，这里就是你放波表的地方了！！如果你用过FM7的话，试着在此构造出那些波型（你可以认为这是本期的作业！）。

现在接着看下面的代码：

```
// make frequency (Hz) table
```

这一段看来是“制作频率表”。

制作频率表是计算出128个MIDI音符对应的频率值的表，根据科学家的推导，每个小二度的频率的比值是常数1.059463094359，这样就可以利用一个循环来计算出这张音符—频率表：

```
double k = 1.059463094359; // 12th root of 2
double a = 6.875;  // a
a *= k; // b
```

这一句表示a跟k相乘的结果再给回a。

比如a=3，k=4时，a *= k就是 3*4。这跟+=, -=是一样的，同样也有/=等等。

```
a *= k; // bb
a *= k; // c, frequency of midi note 0
```

算到这里为止，a的频率就是MIDI中最低那个音符的频率了。

```
for (i = 0; i < kNumFrequencies; i++)  // 128 midi notes
{
  freqtab[i] = (float)a;
```

把a当时的值给freqtab [i] 。比如第一次进来这个循环时，i为0，即freqtab[0]为刚才算到的a值。

```
a *= k;
```

这时a的值改变了，所以下次再进循环时，freqtab[1]的值就已经是相对这次的小二度的值了。依次类推，经过这一轮循环后，128个MIDI音符的频率就算出来了。

```
  }
}
```

这个频率表，几乎可以认为是你的VSTi必需的一个表，所以这段代码你就带上防身吧。

前面是制作波表，这里是制作频率表，到底要制造多少张表我们才能看到VSTi的真谛啊？！

说真的，这里的这些表跟VSTi本身真的没有什么关系，只是为了让大家真正能看懂后面的代码才介绍如此之详细的。在下一期就会避开这一种层次的东西了，所以大家要留心了。

在任何时候，如果一个程序在制作表，特别是在初次运行时（比如本例），那它的最主要的目的就只有一个：节省运算时间！

这是编程界中有名的“用空间换时间”论！

假如告诉你从x变到y可以用一个公式计算出来，但这个公式不仅有加减乘除，还有正弦、余弦、指数和双曲线函数，那么当我问到你的时候，你能很快算出来给我吗？

但如果你事先把可能问到的x及其对应的y值都算好了，并按照x从小到大写在一张表格中，那么当以后被问到的时候，你可以很快找到这个x对应的y值。

像本例的频率表，虽然可以用一个公式算出，但代价是很大的。首先给出其公式：

$$y = a \bullet k^{x+3}$$

该计算中有一个加法，一个幂运算，和一个乘法。而对于C/C++语言来说，一个幂运算要写成一个指数运算、一个对数运算，再免费送一个乘法运算！！所以要做到这个结果，要写成如下的程序：

```
y = a * exp ( log ( k ) * ( x + 3 ) );
```

而如果事先把值算好了，那在用的时候，只要告诉它x是多少，就直接可以从数组中去取就行了。这个时间就相当于计算一个整数加整数的时间。

你几乎可以认为，你用心算、口算、算盘算或者笔算如上的等式的结果所用的时间，跟你计算2＋3的时间，就是电脑用查表法求得结果的时间跟即时去算求得结果的时间的这种差距感！！

前面一直说到数组，那么本例中的几个数组又是具体怎样使用的呢？

首先我们上次说到Delay时用到的buffer就是一个典型的数组，不过它其实是个指针，比本例中的还要复杂一点，但不管是数组还是指针，在使用的时候都是很相似的：

数组名[元素号]

比如上回的buffer[i]，只要改变i值就能读写这个buffer的第i个元素。

0	1	2	3	4	5	6	7	8	9
12	15	-9	3	0	33	190	-99	27	9

假设这个buffer的前10个数如上表所述，那么buffer[4]就是0。

如果有buffer[5]=12，那么下次再读取buffer[5]的时候就能取到12了。

数组的声名跟一般变量不同的是，需要在后面写上它的“最大可能元素个数”，比如本例的，就能找到其定义的地方（寻找的方法同寻找kWaveSize）：

```
static float sawtooth[kWaveSize];
static float pulse[kWaveSize];
static float freqtab[kNumFrequencies];
```

这表示两个波表的元素个数是kWaveSize个，而freqtab(频率表)的则是kNumFrequencies个。从类型上看，这三个元素都是float型的，即浮点型的。

而前面的static则表示这几个数组只能在当前的文件中使用，也就是只能在vstxsynthproc.cpp这个文件中使用，其他文件如果出现以上三个数组，就会报错！

本期讲了很多跟C/C++相关的东西，虽然看上去是好像没有讲什么，不过有了这些基础，下次我们过其他的VSTi接口函数以及处理的过程时就能快很多了，希望大家有C/C++相关的问题要大胆在论坛上问，我会一一解答的。

如果还有其他问题请直接留言到MIDIFAN，或者写信到我的邮箱：seerxillion@126.com

点这里进入论坛讨论此文

从声音到声音——音色合成技巧（二）

如何玩好音高、音量、音色、音长

csounding/文

这次我们将重点讨论合成部分。因为标题是如何玩好音高、音量、音色、音长，所以我们首先按照声音诞生的顺序来研究一下合成器，然后看看怎么样运用这四个因素，了解一下音色制作的另一部分。前面已经说过了，声音的四个要素包括音高、音色、音量、音长，那好吧，我们开始看看合成器究竟用什么手段来实现声音的四个要素，同时也解开“调制”神秘的面纱，毕竟调制这个词已经在模糊的状态下泛滥的使用了，而且这个调制和烹调时候的调制不一样。

对于任何的合成手段，都要有一个最基本的发音的手段，这样才能有一个基本的声音可以用来修饰。如果按照老生常谈，曾经的很久以前，在所谓的模拟合成阶段，要知道什么VCO……挡住挡住挡住！不要这种老生常谈。其实什么VCO，什么DCO，这个O，那个O，这个WG，那个Multisample……其实，这些就是一个能够发声的东西。只有这个东西的存在，合成器才能发出最基本的声音。比如，当我们要开始制作采样音色的时候，我们所需要的能够发声的东西其实就是上一节谈到的采样。知道这个东西的用途以后，我们给这类能发出最基本声音的部分起一个名字，叫振荡器，也就是Oscillator，哈哈，到底还是一个O。所以，在制作音色（不是调制，因为现在调制还沾不上边）时，我们首先要选好或者做好一个基本的声音。

好了！基本的声音出现了，哈哈，不过，一个采样怎么会有音高的变化呢？“那就用Waves的Pitch Shifter变调吧……”这是谁说的？其实手段对了，但是实现的手段也太复杂了点哈！所以合成器的下一个部分该赋予采样来具有音高的变化了，这个就是合成器中音高的部分。我们在振荡器的后面放一个专门用于变调的部分，然后用键位来控制音高（注意），这样，一个采样在键盘上弹得时候，就有正常的音高变化了。知道这里为什么用正常呢？因为在某些情况，例如制作效果类音色，我们并不需要跨度很大的音高变化，倒是微调的音高足够了，这个时候，我们就可以做出不正常的音高变化，比如键盘4个八度对应音高变化的1个八度。这样看来，我们可以用一个参数来控制键盘键位影响一个参数的强度，这个在合成器叫做Key Follow或者叫Key Track，例如Key Follow控制Pitch（注意），当Key Follow为100的时候，就是正常的音高变化，当Key Follow为50的时候，键盘2个八度对应音高的1个八度。好了，现在我们已经认识合成器里面的两个部分了：振荡器，音高。

现在我们可以在键盘上面弹奏一种很难听得音高了，像不像当年的KD49电子琴呢？其实是因为我们能够表达或者控制的声音的要素太少了，那怎么办呢？我

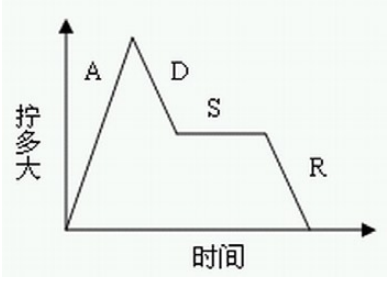
们尝试改变一下音色吧！恩，已经听到有人喊了，滤波器！滤波器！……哈哈，滤波器其实就是一种类似于均衡器的东西，当一个均衡器的Low/High Cut均衡的Q值可调整，那么，就可以叫做Low Cut/High Pass滤波器和High Cut/Low Pass滤波器，而Peak & Dip或Bell的均衡，就是典型的Band Pass/Band Eject滤波器。这时大家应该有一个认识了，既然滤波器就是这样的一个东西，所以滤波器并不能产生出全新的音色，而仅仅是在振荡器产生的声音上进行修饰，这就是为什么采样合成中振荡器也就是原始采样波形的重要性。不过也有比较怪的滤波器，例如E-mu的Z-Plane滤波器，或者01/W中的Wave Shaping，他们都能改变声音原有的谐波成分，因此都可以作为改变音色的手段。不过滤波器在很多文章中都说了很多次了，这里就提一下用途，详细情况可以参考其他文章。

好了好了，现在已经三块了，振荡器，音高，滤波器。现在该是音量大放光芒的时候了，哈哈，音量可能大家没有觉得它在声音中起多大意义，但是在音色制作中也是很关键的，比如我们有一个音色有两个振荡器，当我们交替的改变两个振荡器的音量（注意）的时候，也就制作出一种很有趣的音色。就像上一节提供的那些MP3中的音量对比例子。另外还有，现在厂商总是宣传：我们的键盘是力度响应的！其实，最直观的感觉就是，我们使劲弹，音量也就越大。为什么？是因为击键速度Velocity控制了音量（注意）。（力度怎么跟速度有关系啊？其实要感谢牛顿第二定律，F=ma，而a=v/t，所以F=mv/t，这样击键速度就和力度扯上关系了）

现在我们手上已经有四块了：振荡器，音高，滤波器，音量。这四块就是合成器尤其是采样合成器最基本的部分。每当我们开始制作音色时，就是在这四块中游来游去。下图就是基本的合成器结构：



这四块中，振荡器决定了音色，音高决定了音高，滤波器决定了音色，音量决定了音量。为什么没有提到音长啊？稍等，别着急。现在就道来（嘘，那个吵吵ADSR的安静一下）。很多人一定知道什么ADSR，什么起音，衰减……老生常谈靠边站！其实什么ADSR，学名叫作包络，这个神秘而且迷糊的东西其实就是一个能够按照时间变化的“手”。比如常见的Attack-Decay-Sustain-Release“手”，就是这么拧的：



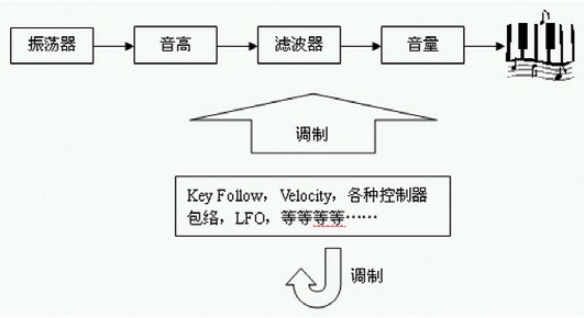
当ADSR这只手拧到音量的时候（注意），那么就是一般说的起音，衰减……但是如果这只手拧到音高（注意）呢？哈哈，就是音高随时间变化。拧滤波器（注意），就是滤波器随时间变化。所以下次说到包络的时候，不要上来就说起音，衰减……因为通常合成器都具有2-3个包络发生器，来控制一定的参数（注意），所以一定要说什么什么包络，比如音量包

络，音高包络，滤波器包络等等，这样说包络才具有实际的意义，同时也不至于改变音量却拧滤波器包络的傻事发生。当然包络还有其他各种的形式，比如创新声卡上面的Delay-A-Hold-D-S-R，KORG常见的A-D-Break-S-R等等，都属于这种随时间变化改变参数的部分。其实音色中的音长概念就是通过这种形式体现的。

人有两只手，合成器也不例外，除了包络这个手以外，还有一只手，就是LFO，学名低频振荡器，什么锯齿波，三角波……老生常谈靠边站！其实这个LFO就是一只能够周期性改变一个参数（注意）的手。而具体改变大小的方法和速度就是LFO的参数。同样因为文章很多，同时在实践中改变LFO的参数变化比较明显，所以大家可以参考其他文章，或在下次的实践中了解，这里只解释其用途。

哎呀，有人不服气了！“我也有手，自己拧不行么？”哈哈，当然可以，除了包络，LFO以外，很多琴和MIDI键盘都提供了各种的手可以直接拧到的控制器，比如弯音轮，这个轮子控制了音高（注意），叫做Cutoff的旋钮发送74号控制器控制滤波器的Cutoff（注意）等等等等，这些都能够直接的控制一定的参数（注意），同时，这些控制手段也可以互相控制，所以我们现在把这些部分加上，就形成了一个比较完整的基础合成器。

在我们展示图之前，大家可能会注意到我在文章的很多地方加上了这样的标记：（注意）。这些个标记有“注意”的地方其实就是大名鼎鼎的“调制”。所谓调制，其实就是用一个参数来控制改变另一个参数，例如那个叫做Modulation Wheel的调制轮，为啥一推音就颤了呢？我们现在仔细分析一下：音颤了，说明音高发生的改变，同时颤字说明了音高是周期性变化的，一推说明那个轮控制了周期性控制音高变化的强度。好了，三个因素分析完毕后，我们知道了这个轮实现的整个过程：调制轮控制了LFO控制音高的量。这正是调制的内涵所在。所以，我们可以通过把一句话中的“调制”和“控制”互相调换来检查其使用“调制”这个词的正确性。比如上面的句子：调制轮控制了LFO控制音高的量，说成调制轮调制了LFO控制音高的量，意思没有变，正确。因为击键速度Velocity控制了音量，说成因为击键速度Velocity调制了音量，也可以说通，没有问题。但是最常用的说法：这是我“调制”出来的音色，说成这是我“控制”出来的音色。莫非这个音色一定要使用现在进行时才有意义。这就说明使用“调制”存在问题。往往这种情况，用制作或合成更合适。好了，调制的身份澄清后就可以把合成器的基础结构图给大家看了。



好了，了解了合成器的神秘，现在应该知道制作音色的一种基本的手续：首先完成基本的振荡器制作，然后完成音高、滤波器、音量的基本调整，这时音色已经有了基本的雏形，然后再通过调整包络等

等修饰部分，来完成音色的制作。

对于理想的合成器，什么参数控制什么参数都是完全可以自定义的，这正是古老的模块合成器的结构，而现在的合成器往往给定了固定的调制方式，因此在音色制作中会有一些限制，正因为如此，比较灵活的合成器都提供了一些完全可自定义的调制手段，例如Emu的Patchchord、YAMAHA XG的AC1、KORG的AMS、Roland和其他厂家的Modulation Matrix概念等部分，或者Kurzweil几乎能够用任何参数来控制任何参数等等来增加合成器的灵活性。在软件方面，Kontakt做的比较灵活，这也是我选择用Kontakt作为实例的原因。

如何玩好这些部分也属于一种创作，从而制作出不同的音色。按照难度可以分成以下情况：

1.用一个参数同时控制多个参数。例如当力度仅仅控制音量，则音色在各种力度下并没有什么变化，音色整体显得很死板，如果此时力度同时控制滤波器的截止频率，则不同的力度下会产生音色上的细微变化。如果在现在的基础上再加上音量包络的控制，那么不同力度下也会造成声音更多的细微差别，这样一个音色就从死板变得灵活了。（对比MP3：1-1 Vel2EG Test 1与1-1 Vel2EG Test 2，本例并无任何力度分层）

2.用一个参数叠加一个参数控制另一个参数。其实有两个例子大家都遇到过，但是并没有意识到原理，其中第一个例子就是上面提到的调制轮。第二个例子则是它旁边的弯音轮：弯音轮配合RPN控制音高变化。弯音轮在-8191－8192的范围控制弯音的强度，而用RPN来控制弯音的范围，结果就是强度控制了音高变化的范围。当然对于自己制作的音色，也可以创造一些比较独特的手段，比如上面提到的用力度控制滤波器截止频率，如果用力度控制滤波器包络到滤波器截止频率的强度，根据合成器的不同，则会使滤波器包络控制滤波器截止频率的强度发生更合适的变化，因为原来叠加的控制量会变成乘法的控制量，相当于提高了控制精度，而不是简单的增加控制量。（MP3：2 Vel2EGInt，本例并无任何力度分层）

3.打破常规的合成习惯。虽然我们提过制作音色的时候首先要按照音色的需要选好合适的采样，但是那些都是对于制作常规音色。如果制作效果或铺垫类音色，我们往往可以尝试打破常规的合成手段，例如制作铺垫音色，我们可以选择常见的电钢琴音色，然后将音量包络的起音和释放延长，就可以作出一个具有电钢琴风格的铺垫音色。类似的，我们可以用常规乐器的采样搭配特殊的包络、LFO等手段来制作各种奇怪的音色。（对比MP3：3-1 E.Piano Original与3-2 E.Piano Pad，本例只改动过音量包络的A、R段时间）

上一次大家应该听一些例子了，这次呢，可以进一步总结上次听得各种例子，上次主要听什么发生了变化，这次可以将重点移到怎么发生了变化，例如周期性还是用包络还是什么其他方法，下次我们将通过实际的音色了解这些部分的应用。

M-AUDIO®



NEW 全新上市

MidAir 25 "无限25"

25键无线MIDI键盘控制器

从现在开始，M-Audio “无限25”让你摆脱电缆的束缚，自由释放你的演奏激情！

- 25键键速响应无线MIDI键盘
- 8个可分配的旋钮，可以在舞台或录音棚里遥控音频软件
- 接收器通过USB与电脑连接，具有MIDI输入输出接口，可用于控制软件和传统MIDI设备
- Windows XP和Mac OS X即插即用，无需安装驱动
- 无线传输方式：2.4 GHz无线电
- MIDI键盘通过9V电源适配器或电池供电



MidAir 25无线接收器

怡生防伪 谨防假冒

EASTERNEDISON 怡生飛揚

独家总代理厂牌：M-AUDIO Toft Audio Designs Brent Averill JOEMEEK FRONTIER DESIGN GROUP ableton

总代理厂牌：digidesign GT POWER MOTU creamware™

欢迎各省市经销商与我们联系，详细资料请访问：http://www.easternedison.com E-mail: dealer@easternedison.com

地址:北京朝阳区朝外小庄6号中国第一商城丹佛豪园16B 邮编:100026 电话:010-65860065 传真:010-65860095

萬寶(國際)音響有限公司

AKAI PROFESSIONAL





MPC 2500

功能強大

舞曲採樣器 / 編曲機 / 音序器







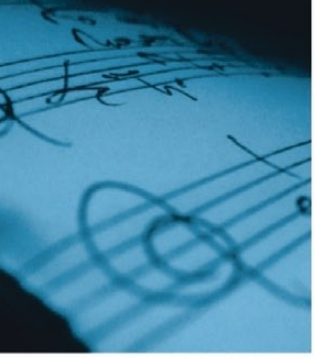


MPC 1000

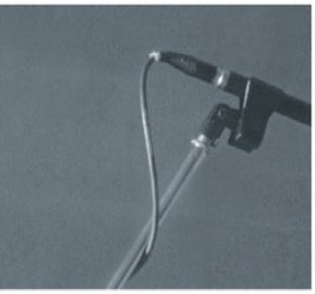
美国『雅佳』专业

AKAI professional

www.marlboroaudio.com



出色TR 出众表现


















请浏览actonchina.com有关其详细规格

从伟大的想法到出色的音乐，既需要创作灵感，也离不开实力雄厚的工作站。新一代高灵敏的、超干劲的KORG Tr音乐工作站，可让你的音乐演奏非同凡响。

TR备有KORG超级效果引擎的HI声音引擎，64 Mb记录容量，扩展了样本组，增加了多种乐器的声音，512个音效程序，128GM容量的声音和9样EM鼓声，通过TR的USB MIDI端口，SD储存插槽及选配的采样升级，使好物共享，与Triton源于一处，音效及声音相互通用。

它黑色实体，友好界面设计，完美的音响成为无数音乐家灵感的源泉，这就是我们隆重登场的黑美人TR音乐工作站

TR
MUSIC WORKSTATION



中国总经销

acton
雅登音响乐器(上海)有限公司

中国总部
地址：上海市浦东新区
唐镇王港虹叶路9号
电话：86-21-5858 9280
传真：86-21-5868 0531
网站：www.actonchina.com
电邮：enquiry@acton-sh.com

金圆办事处
上海办事处
电话：86-21-5858 9280
传真：86-21-5869 0531
北京办事处
电话：86-10-67020066
传真：86-10-67019993

广州办事处
电话：86-20-8339 7731
传真：86-20-8339 7428
成都办事处
电话：86-28-8522 7631
传真：86-28-8522 7632

中国陈列室

TOM LEE Music 通利琴行

通利琴行北京旗舰店
地址：北京市崇文区天坛路57号
电话：010-67020099

深圳：(755) 8329 3828　上海：(021) 5228 2065
广州：(020) 8386 4835　宁波：(574) 8731 3256

数字音频领域最优秀的专家

为您提供专业的服务

选择爱新 聚福未来

爱新聚福

www.aixinmusic.com

我们只想做的更好



优惠的数码音乐器材销售





专业的数码音乐技术支持
大中小型录音棚工程建设



国际数码音乐器材一级零售商

爱新聚福

展 厅：北京市海淀区中关村鼎好电子大厦 4999
办 公：北京市海淀区中关村鼎好电子大厦 811
电 话：(010)82697874　82697832　82697849
传 真：(010)82697849　手　机：1361256520
E_mail:aixin@aixinmusic.com



“＋中音数码学院”电脑音乐制作课程培训内容简介

众所周知，现代流行音乐的发展总是与电子技术的不断发展紧密相连，在电子音乐领域最引人注目的成就应首推MIDI技术的产生、发展以及多方面的应用，越来越多的人热衷于钻研电脑音乐制作技术，又苦于没有专业的、系统的学习环境，由此“＋中音数码学院”应运而生，“＋中音数码学院”以中音免维护音乐电脑为核心，配备专业的MIDI键盘、音频卡、音源、监听音箱等设备，由具有丰富教学经验并富有创新精神的专职讲师任教，通过讲师的示范和讲解将可以学习到各种软硬件设备的操作及其环境设置。“＋中音数码学院”的培训宗旨：面向社会，针对不同水平的学习者，开设专业系统课程，使学习者在电脑音乐制作技术和录音工程技术方面，成为基础理论扎实、知识结构完善、具有娴熟操作技巧的多元化的音乐制作人，帮助您早日实现从事电脑音乐制作的愿望。

本期电脑音乐制作速成培训班将于暑期开课，教学内容包括电脑作曲、编曲和多轨录音。不用担心自己的音乐基础，您将迅速学习到与世界同步的Midi及电脑音乐方面的基础知识，硬件方面包括电子合成器，音源，采样器，电脑音乐系统等现代电脑音乐个人录音棚中各种设备的使用方法，软件方面包括作曲及编曲软件Cakewalk和 Sonar系列软件的详细使用，多轨数字录音软件Cubase、Samplitude 和CoolEdit的详细使用方法，Giga等软件采样器、软件合成器的使用及各种吉他软件、打谱软件、自动伴奏软件、练耳软件及音乐学习软件的使用，为您发挥自己的创作灵感，创作和制作优秀的音乐提供坚实的保证。另外还可以学习到如何使电脑音乐系统与视频设备的同步、如何制作影视配乐、广告音乐等。使您成为全面的音乐制作人。其他特别增加了乐队和录音棚方面的知识，包括Midi吉他、电鼓、Midi小提琴等各种真实乐器控制电脑音乐系统的方法和编曲法，各种软件效果器，插件效果器安装及使用，人声、吉他、贝斯、鼓及乐队的录音制作及缩混方法。使您迅速成为专业的录音师和全能的音乐人、乐手。

现代音乐要求音乐人具有与世界先进水平同步的音乐制作理论和知识，这样才能使您的音乐理想可以插上翅膀，在天空中飞翔。如果您有音乐的一个梦，有很多创作灵感，为何不付出努力让他成为现实？！

公司名称：北京海嘉欣韵电子设备有限责任公司
联系电话：6707 5581 6733 9932
公司网站：www.netpcmidi.com
公司地点：北京崇文区珠市口东大街2号3层30号“＋中音数码学院”海嘉示范中心



KORG 编曲机 让你更专业

特价套装 送专业及中国民乐风格磁盘

买

Pa1X/Pa1X Pro
高创造性的编曲工作器

- RX真实体验(Real eXperience)编曲引擎
- ROM内存有850种声音，包括一个立体声声学钢琴，加上48种鼓套全新的鼓声采样`全新的贝斯采样`全新的吉他声`全新的萨克斯`全新的管风琴(旋转风琴和教堂风琴)`全新的管乐(独奏和合奏)
- TC-Helicon技术的人声效果器
- XDS(Dual Sequencer)双音序器设计,一个音序器演奏时另外一个音序器可以进行数据装载
- 可编程的音乐数据库的歌曲书Songbook
- Pa1Xpro: 76键半配重键盘，8个数码拉杆控制 ,240×320色彩的全彩色TouchView可接触式的显示屏
- Pa1x:61个琴键,4个扬声器双功放系统

送

GATOR 专业带滑轮键盘机箱



+

AKG K44专业耳机



GATOR 专业带滑轮键盘袋（PA80）



买

PA50/PA80
出色的声音，便捷的操作

- TRITON合成器引擎
- 超过660种音色及32套鼓组， 128种用户音色及64种用户套鼓音色
- 音乐风格可存储304种音乐风格（预装有256种）
- 音乐风格要素2个前奏/4个变奏/2个过场/2个尾奏/淡入淡出 Pa80独有“后台音序器”，在单通路中多达8个音轨的风格伴奏以及8个音轨的实时播放
- Pa50具有2个15W的功放和2个安装于反射音箱中的扬声器;Pa80内部声音系统则有22W的立体声功放和4个扬声器。
- Pa80具有Pa50的全部功能，并拥有更强的主机、更大的容量以及更多的用户可选配设备，以满足更加专业的需求。

送

Warwick 专业键盘袋（PA50）



+

AKG K44专业耳机



中国总经销

Acton 雅登音响乐器(上海)有限公司

中国总部
地址：上海市浦东新区唐镇王港虹叶路3号
电话：86-21-5858 9280
传真：86-21-5868 0531
网站：www.actonchina.com
电邮：enquiry@acton-sh.com

全国办事处
上海办事处 电话：86-21-5858 9280
北京办事处 电话：86-10-67020066
广州办事处 电话：86-20-8339 7731
成都办事处 电话：86-28-8522 7631

中国陈列室

TOM LEE Music 通利琴行

通利琴行北京旗舰店
地址：北京市崇文区天坛路57号
电话：010-67020099

深圳：(755) 8329 3828 上海：(021) 5228 2065
广州：(020) 8386 4038 宁波：(574) 8731 3256

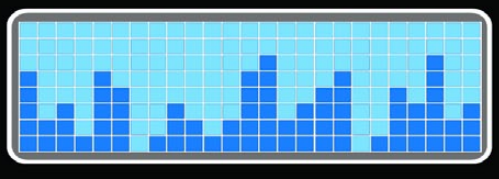
有关KORG PA系列特价套餐信息请与雅登各办事处或 经销商查询，详细规格请浏览www.actonchina.com

中国数字音频电子商务走向何方？

中国数字音频行业的黑马

www.ddYS.com.cn

大地原声数字音频网



倡导音乐创作革命 引领数字音乐先锋

中国最专业的ROLAND乐器论坛

ADD：厦门市思明区西堤别墅80号 TEL:0592-5331939 5333083

搜狐数字艺术
SOHU Digital Arts

创意生活 快乐分享

首页 ···› 数码天下 ···› 数字艺术

资讯
教程
赏析
博客
社区
.....



点这里进入论坛讨论此文

把USB线和电源线咔嚓掉
——M-Audio MidAir25 无线MIDI键盘评测

musiXboy/文



MidAir25 包装

今日收到怡生公司送来的M-Audio MidAir25无线MIDI键盘。这可能算是M-Audio历史上最急茬的硬件产品，从2006年6月3日正式发布，再到8月初全球上市并进入中国市场只用了不到2个月的时间。

如此快的动作，说明了M-Audio历史上第一台无线MIDI键盘的研制并不困难，另外就是CME在2006年1月冬季NAMM展会展示了无线MIDI接口后，M-Audio坐不住了，他们要抢先发布民用市场的无线MIDI设备。

第一印象：

打开包装，里面除了MIDI键盘和电源、USB线以外，多了一个无线的MIDI接收器，外型设计的确实非常漂亮。



无线的MIDI接收器



MidAir25 键盘

在背面可以看到写着无线技术由Frontier Design公司提供授权，原来M-Audio是买下了Frontier Design公司的无线技术，再结合自家的老Oxygen 8键盘，装配出了这一整套的MidAir25无线系统。关于Frontier Design公司的无线技术，请看我以前写的《Frontier TranzPort 无线你的无限》一文。

体验无线：



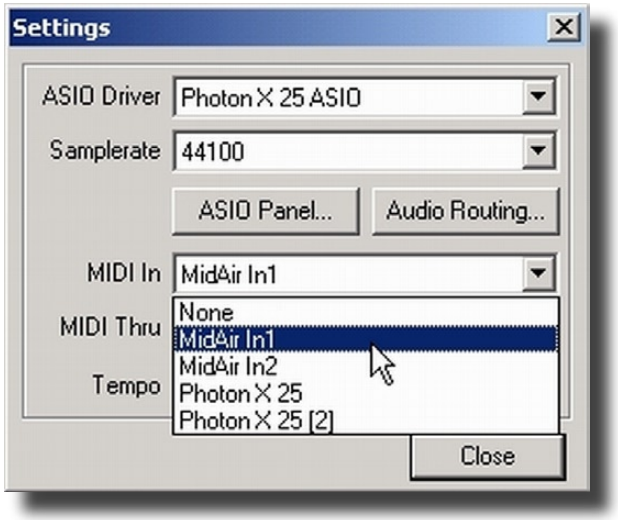
电脑——无线MIDI接收器——无线MIDI键盘

首先把无线MIDI接收器连接到电脑的USB接口，只要使用的是Windows XP或Mac OS X系统，都不需要驱动。

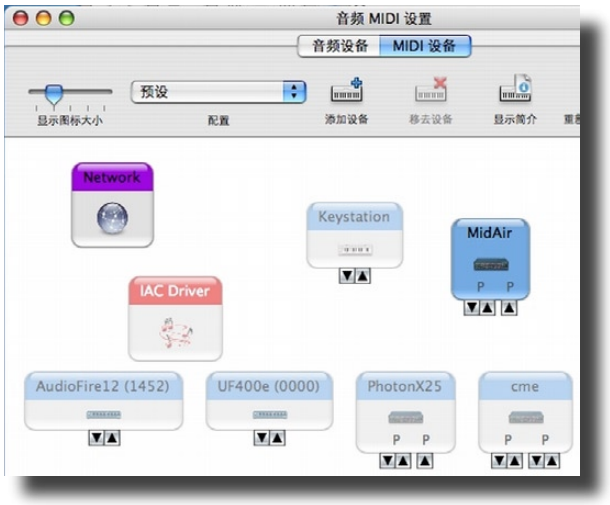
再拿出MIDI键盘，虽然现在可以直接无线发射MIDI信号，但不供电还是不成

的，别指望电源也能通过无线方式获取。唯一的解决办法就是使用6节5号电池（当然你也可以用电源线直接为键盘供电，但那样就又退化成“有线”键盘了，失去了无线的意义）

把无线MIDI接收器连接到电脑，再把6节5号电池塞入键盘，在电脑里就可以看到MidAir 25键盘了，你就能抱着键盘随心所欲的演奏了。



在Window XP下认出的MidAir 25无线键盘



在Mac OS X下认出的MidAir 25无线键盘

无线抗干扰：

介绍中说MidAir 25使用的是2.4G这个频率的自由无线信号，我们手机上用的蓝牙无线传输也在这个频段。所以我特意在演奏MidAir 25的时候，打开手机，与电脑用蓝牙的无线方式传输了一张图片。一切OK，图片正常传输，键盘也正常演奏，没有任何冲突。

以前使用Kenton的无线MIDI接口，会遇到打手机就有干扰的情况。这次我特意试了一下MidAir 25，在演奏键盘的同时：

- 拨打手机
- 拨打小灵通
- 蓝牙传输文件
- 手机无线上网
- 通过802.11g家庭无线网络上网

都没有任何干扰，表现很好。

无线距离和穿透性：

说明书上介绍MidAir 25的无线传输距离是10米，键盘和接收器的距离超过10米，就玩不转了。我在电脑前演奏键盘完全没有问题，在开着门的情况下，径直走出房间5米也没问题。但拐弯的话情况就不太妙了，因为无线信号不会拐弯，它需要穿墙，此时的无线距离会缩短。关上门之后可响应的距离就缩的更短了。对比我之前用过的Froniter的Tranzport无线控制器，MidAir 25似乎比它的响应距离要稍微短一点点。

不过我想不管是在工作室里，还是

在现场，你都不会隔着一堵墙去演奏键盘吧？在无阻挡物的情况下，MidAir 25的无线距离还是足够我们用的了。

细看无线接收器：

虽然叫无线接收器，但实际可以看到它还是有1进1出的普通MIDI接口，你可以当作一般的USB MIDI接口使用。



无线接收器可以当作1进1出的MIDI接口使用

在侧面，有一个模式转换开关，一般放在auto模式下。此时无线接收器能接收来自MidAir 25键盘的MIDI信息，本身自己又能作为一个1进1出的MIDI接口使用，通过有线的方式连接音源和第2台MIDI键盘。



无线接收器的三种模式

细看MidAir 25键盘：



MidAir 25与老Oxygen 8

MidAir 25几乎就是改良版的老Oxygen 8，先看个大对比：

	老Oxygen 8	新MidAir 25
键盘	25键	25键
音高区域增减按钮	有	有
弯音轮	有	有
调治轮	有	有
旋钮	8个	8个
推子	1个	1个
按钮	无	1个
显示屏幕	3位数字	3位数字
走带控制	无	无
电池供电	有	有
功能按钮	1个	1个
延音踏板接口	有	有
MIDI输出接口	2个（USB输出和键盘输出）	2个（无线输出和键盘输出）
MIDI输入接口	无	无
开关	无	有
音色切换按钮	无	有
切换键盘音高区域	可以	可以
存储控制器参数	不可以	10个
快照（snap-shot）功能	无	无
Windows XP自动识别	不可以	可以
Mac OS X自动识别	可以	可以

主要改进其实就在于多出了选择音色号的+/-按钮，通过这两个按钮发送音色改变信息，边演奏边随时换音色。老Oxygen 8没

有专门的换音色按钮，换音色必须先按MIDI Select，再按Program键，再按+/-号，麻烦。



MidAir 25选择音色号的+/-按钮

MidAir 25可以在琴上直接存储10组控制器参数，随时调用。老Oxygen 8一组也存不了。



Sustain按钮和无线连接指示灯



推子和显示屏

另外MidAir 25多出了Sustain按钮，它的功能和踩下延音踏板一样，使按下的音符保持下去而不必一直按着琴键。这个灵感估计来自于一些模拟合成器上的latch键，可以在输入旋律以后到软件里调节音色，还是很有实用价值的。

其他方面MidAir 25与老Oxygen 8几乎是一模一样，包括键感。

总结：

MidAir 25我个人感觉推出有些过于仓促，把Frontier Design的技术买过来，内置到一个MIDI键盘里就推出了。这个无线接收器只能接收无线MIDI信息，而不能无线发送MIDI信息。为什么不把那个无线发送MIDI信息的模块单独拿出来销售呢？

其实目前不少人家里都已经有MIDI键盘了，他们不需要一个还不如自己新买的M-Audio Axiom的无线键盘，而只需要一个无线MIDI接受/发送器，能够让已有的MIDI设备都拥有无线传输能力就很好了。

现在还没购买MIDI键盘的朋友，25键或37键的键盘可以考虑MidAir 25/37，毕竟无线是未来的趋势。如果你喜欢抱着键盘在舞台飞奔，在工作室里跑来跑去找灵感，那就不要考虑功能多而全的键盘，你需要一个无线键盘。

不过我认为M-Audio下一步会把MidAir 25琴里的无线MIDI发射器单独拿出来，与无线MIDI接收器配套销售，满足大家将已有设备无线化的迫切需求。其实M-Audio已经准备这么做了，官方透露他们在最近会单独销售无线接收/发送器，不过价格还没公布，想玩无线的朋友可以开始攒钱了。F

点这里进入论坛讨论此文

体验SSL魅力↘

大觉者/文

SSL 4000 Collection效果器插件

提起英国SSL公司的模拟调音台，那可是让所有喜欢玩音频的人垂涎三尺的东东。它稳定的性能、效果品质以及卓越的音质都可以说是无与伦比的，可谓音频设备中的名器。国内外很多顶级的录音棚、直播间都在使用SSL的产品，能够用上这个东西可以说是所有音乐人的愿望。



图：真正的SSL调音台

当然，这家伙百万元的价格也不是一般人能“享受”起的，呵呵，不过，在这个插件的时代，已经没有什么是不可能的了，越来越多的经典硬件都在不断软件化，SSL也不例外，最近，插件再一次施展它的魅力，使SSL这只“旧时王谢堂前燕”也飞入了寻常百姓家。让更多的人都能够在电脑上体会到SSL效果器的优秀品质。在这篇文章中，我们就一起来体会一下SSL的效果器插件吧。

首先要说的这套插件，是由大家耳熟能详的Waves公司出品的SSL 4000 Collection，它是Waves与SSL（Solid State Logic）共同合作的，一共包含三个SSL 4000的经典效果器。分别为G-Master Buss Compressor（压缩） G-EQ（均衡器）和E-Channel（通道条）。

SSL 4000 Collection所包含的3个效果器，是Waves和SSL的工程师一起合作的，完全参照SSL 4000的硬件制作，Waves的工程师花了一年时间进行物理建模等分析和设计，最终使得这套插件的声音与SLL 4000硬件几乎完全相似。当然了，只是相似，并不是完全没有差别（那谁还买硬件啊），不过，这也已经相当不错了。



图：Waves出品的SSL4000 Collection效果器套装

据说，为了证明它处理出的声音和真正的SSL一样，Waves曾进行了一个测试，将相同信号由SSL 4000硬件和SSL 4000 Collection同时处理并反相进行播放，结果产生了35dB的相位抵消！说明这个软件和硬件的声音确实有着非常多的一致之处。

SSL4000 Collection安装上之后，和其他Waves的效果器都排列在一起。如图：



图：选择SSL插件

先来加载一个压缩看看。这个压缩器的界面做得非常精致漂亮，完全是仿照硬件来设计的。操作也比较简单，常用的旋钮诸如THRESHOLD、ATTACK等都是模仿硬件设计的，使用起来非常方便，就像操作硬件一样。



图：SSL压缩插件

它的声音非常有特点，感觉模拟味道“浓郁”而又细腻。我真的是非常喜欢这个压缩的声音！以前我非常喜欢“蓝色电子管”的那个压缩，如果和这个比，两者风格上稍微有点类似，但SSL明显比那个要细腻多了，声音要更干净利落的多。

无论是高速、瞬间的压缩处理如打击乐，还是缓慢而长时间的细腻处理如钢琴、弦乐，它都能很好地胜任，且不迟钝，反映很灵敏，用在打击乐上的时候，几乎每个爆发点它处理得都很好。不过它的预置选项很少。当然它本身就是按照一个硬件那样来设计的，更适合像一个硬件那样用来自己调声音。



图：SSL压缩的预置

这个压缩带有增益补偿，另外还带有一个很独特的自动淡化（FADE）功能，淡化时间可以随意调整。其物理建模功能也可以随意关闭或者打开。

SSL压缩占用的资源非常少，大概和C4或者L2差不多吧。用起来绝对舒心。

接下来我们来看看E-Channel插件。其实，拥有它就是等于你拥有了一个SSL调音台的一小条儿。呵呵，知足吧，真的SSL台子就是一小条偶也买不起啊……唉。

从这个东东的界面上我们可以看到它大致分为滤波、动态和均衡三部分的处理。



图：E-Channel插件

E-Channel插件是参考SSL在1983年与George Martin一起开发的非常著名的Black Knob来制作的。看来还是硬件经典啊！1983年的东西至今照样还是最好的东西……

实际应用中，这个E-Channel插件确实是个很不错的综合效果器，尤其是它的滤波确实是非常不错的。如果你的音乐中低频太厚重或者高频发出吱吱的讨厌声音，它就是很好的一个帮手了，而且无论你怎么调，声音都在最大程度上保证不失真。它的均衡曲线也都是非常平滑的，通路设置更是非常人性化。

E-Channel插件的动态处理部分也很经典，它有一个压缩，一个限制，而且都是支持软拐的！另外，它的门限/扩展完全是按照经典的SSL LS611E进行建模设计的。动态处理部分带有自动补偿增益，保证可以得到稳定的输出电平。无论从哪一点上看，它都更像一个真正的模拟设备，就像是你在操作一个真正的SSL台子，当然，只是一部分……呵呵。说实话我有一种这样的感觉，如果Nuendo本身的调音台每一轨都带着这玩意，那该多……

这个通道条插件的预置还是比较丰富的，可以选择已经调好了的各种预置来适用于不同的声音处理。比如打击乐、吉他、弦乐、钢琴、人声等。



图：通道条的预置

最后一个插件是一个均衡器。可别小瞧这个小均衡插件，它可是完全模仿SSL G 系列经典的 EQ292制作的。完美再现了SSL均衡器的声音。这个均衡是4段的，比通道条上的均衡要更精细，尤其是增益非常好，同时它的滤波器依然是非常完美。不用说普通的单独乐器或者人声，即使是母带也没有问题。不过，因为它是完全模仿硬件制作，所以用惯了“点线”式均衡插件之后可能并不习惯这种，但如果你更喜欢硬件的那种操作感觉，那么这种均衡操作起来还是很直观的。



图：SSL EQ

同样这个EQ也带有大量的预置可以直接应用于各种不同声音的处理。



图：均衡预置



图：Waves SSL的Logo画面

关于Waves的这套SSL插件就说这些啦。不过，无独有偶，下面我们接着再看一个插件，你一定会感到惊讶。嘿嘿……



图：运行在Scope平台上的SSL压缩

看到这个你是不是觉得很面熟？哈，何止面熟，我们到Nuendo里把它们放在一起瞧瞧：



图：VST版和Scope版的SSL压缩，好像双胞胎

呵呵，这是SLL插件的Scope版本。是digitalaudiosoft为Creamware精心制作的。运行在Scope平台下和XTC模式中，使用它的前提是你必须要拥有Creamware系列的音频卡。比如ScopeHome、Scope Project等。

我们可以看到两者的基本功能几乎没有什么区别。只是运算的载体不一样。Waves的VST格式插件是要靠电脑本身的CPU来运算的，而Scope插件则是使用DSP卡本身的芯片来运算，不消耗任何电脑资源。那么或许你要问，两者的声音有什么区别吗？

其实，关于这个问题说法不一。很多人都认为，DSP芯片和CPU一样都是运算，都是1＋1＝2，本质上没有任何区别。但也有很多人认为，DSP采用的算法和CPU并不相同，而且，DSP完全用于音频处理运算，其插件的品质一般也要更好。我反复对比了两个插件的声音，它们的声音极其相似，但是，我还是觉得DSP版本的声音要好过VST，要么是心理作用，要么就确实如此。总之两者还是有差别的，因为它们并不是一个公司制作的。

除了压缩之外，Scope平台上也有一个通道条的插件，同样具备滤波、动态和均衡三部分的调节。



图：Scope版的SSL通道条

除了这两个版本之外， SSL插件还有另外一个版本呢，那就是SSLDuende版本。这是SSL公司自己出品的。也是DSP效果器。它是一个外置的DSP处理器，通过火线连接电脑之后，就可以使用它来运算自带的效果器插件，同样也不占用任何CPU资源。大家看看，也是一样的东东。呵呵这真是三胞胎了。



图：SSLDuende版本的压缩



图：SSLDuende版本的通道条



图：Solid State Logic Duende外置DSP效果处理器

要使用SSLDuende，就得买这台机器了，价格当然也比较贵，要万元左右，但是，总比上百万元的SSL模拟台子要便宜多了呀！你看，插件时代给音乐人带来的惊喜真的是太多了。无数音乐人梦寐以求的SSL台子，现在大家也都同样能够享用了。这在以前是根本不可想象的，而插件的时代将这些都变成了现实！

Guitar Rig 2 vs. AmpliTube 2

模块化吉他效果器软件的对决

门子/文

随着软件的时兴，不但传统的音源，效果器已经越来越接近硬件，连吉他效果器也不能幸免。通过一个软件，可以让电脑变成一个硬件吉他效果器一样的玩意，这实在是太好了！当然，由于电脑数字运算的问题，想用软件吉他效果器与全模拟电路的单块或者箱头对比是不现实的。不过可惜就算只和合成效果器比，AmpliTube 1和Guitar Rig 1好象也只能达到一些中流吉他合成效果器的效果，不过大家依然对软件吉他效果器抱有非常大的希望。确实，现在通过建模，采样技术，软件效果器已经和硬件差距越来越近了。那么新一代的软件吉他效果器是否有崭新的面貌呢？首先ni公司出品了Guitar Rig2。作为Guitar Rig的升级产品，与1一样使用了硬件踏板与软件计算的结合。紧随其后的AmpliTube2也开发了踏板（可惜要单买）。二者越来越有些争个你死我活的感觉了。那么到底哪个比较好呢？

今天，托midifan老大的福，门子同时拿到了Guitar Rig2和AmpliTube 2。其中Guitar Rig 2是我买的正版，AmpliTube2是演示版（我是NI迷，所以掏钱买了Guitar Rig2）。看起来用一个正式版和演示版比不是很公平。不过好在AmpliTube 2的演示版在声音上和正式版并没有什么区别，而只是在模块上加入了一些限制。所以在下文的评测中，门子会尽量使用两个软件所共有的模块来进行对比。



AmpliTube 2软件：

点这里下载我用AmpliTube 2做的一段声音。

首先我们先介绍一下在国内大名鼎鼎的，解决很多穷吉他手实际问题的AmpliTube的第2代。可以说相比于Guitar Rig 2，AmpliTube2 算是变化非常之大了！连界面都完全变了。

AmpliTube2 响应时代号召，已经从1代一个简单的音箱模拟变成了一个很完美的模块化效果器，它分为5个模块体系：

1．校音器——这个就不用说了吧……

2．单块模拟——所有知名的单块，

你就找吧（可惜demo里面好象少了一些？）

3．音箱箱头模拟——个人认为是amp2最棒的部分。

4．箱体模拟——与Guitar Rig惊人的相似。

5．机架效果器——录完吉他之后的混响，压缩等等都可以在此调整，个人认为是amp2最失败的地方……



刚刚进入AmpliTube2，一上来默认就是校音器部分。虽然这没什么可说的，但它确实好漂亮啊！和Guitar Rig都是小方快的设计完全不同，十分前卫的感觉。



这是……某家装公司的广告么？呵呵，这是单块模拟部分的界面。同样相当漂亮。在这个“木地板”上我们可以放置一共8个单块。对于一般的吉他音色来说是绝对足够了。单块的种类与现在的硬件效果器差不多：失真类，eq类，mod类，delay类。几乎都是模拟了不同时代的经典效果器。虽然名字不一样，但是都可以从界面上明显看出这些效果器的“本尊”。呵呵，你能认出这些失真效果器么？



失真模块



调制类



自由组合

可以说，这些单块效果器的质量是很高的，呆会与音箱模拟来配合使用，质感非常强！

调吉他音色光有单块可不行，还必须有音箱才好。与1不同，AmpliTube2模拟了各种知名音箱和箱体，而且允许用户自己组合~我们先看看箱头部分：



AmpliTube2把箱头模拟拆成了3份以方便用户的调节自由度。分别为前级，eq和后级部分。虽然种类不如Guitar Rig多，但是个人认为它们的质量比Guitar Rig厉害的多！用户可以自由地组合来改变音色。试过用mesa的后级推marshall吗？呵呵。

可以说这是amp2最吸引我的地方了。声音经过amp2的箱头会变的相当舒服，扎实和有质感。闷音，切音表现都非常好！

再然后就是它的箱体模拟了：



相比于Guitar Rig，这个箱体模拟就差的远了。无非就是几乘几的箱体选择，mic选择，off axis ，on axis，远近选择。只能说及格，但说不上优秀。其实经过箱头模拟的声音也是非常舒服的，但是你会发现无论你怎么调整这些参数，都不是非常像真实中mic的调整。声音依然有质感却出不来太多的变化。（有意思的是，mic的图象会跟着你的调整发生变化，十分形象，不得不佩服IK的美工。）



经典sm57模拟，远场拾音……



换箱体啦！

箱头之后，就是8个机架位，我们可以加入一些amp2自带的传统效果器，如压缩，

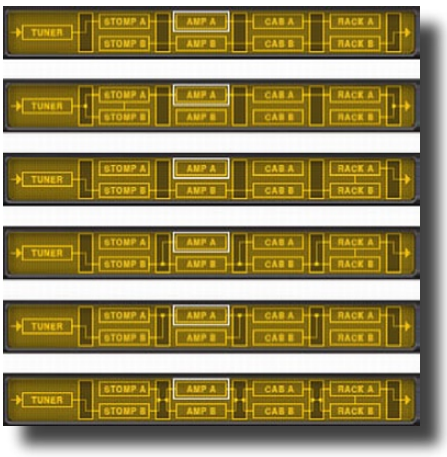
混响等等。我说实话，这几个效果器质量非常一般……与单块那里的东西不是一个档次的东西。



位置是随便排列的。



以上每个模块都是可以同时调用两个的。通过信号路由可以选择它们之间的连接。以此实现模块之间的组合。比AmpliTube1要先进的多。但是完全败给Guitar Rig的是，模块与模块之间的连接并不是自由的，而是AmpliTube2提供几种选择，我们只能选择而不能自己随意调节。



比如我们想在箱头后面接个机架效果器？不行。因为模式已经定了，没有我们需要的模式。当然一般的音色调整已经囊括在这些选择中了。只是有一天需要这样调节的时候，恐怕就有点郁闷了~

总的来说，AmpliTube2是个非常令我惊讶的软件！从音色的可调性到音色质量都是非常好的！任何人都可以轻易做出很好的音色。它的自由度相比于1代也是提高很多的，可是我觉得作为一个吉他软件效果器，它的自由度依然不够。在demo版使用的几天中，我发现想调出一个好音色很容易，但想做一些有个性的音色，却显得有些麻烦了。而且，虽然它的箱模质量很好，但是感觉有些……奇怪……怎么说呢，虽然音色的质量很好，但实际调节起来以往用mic录吉他的经验却用不太上，既是说mic模拟好象不是很“像”，而是很“好”。

Guitar Rig硬件：

大家对Guitar Rig软件很熟悉了，但也许都没看过Guitar Rig 2的硬件声卡+控制器。非常可惜的是AmpliTube 2的硬件声卡+控制器还没上市，所以无法进行对比了。



Guitar Rig 2包装



Guitar Rig 2
说明书和安装光盘



Guitar Rig 2硬件
是声卡+控制器的混合体



背面接口，可同时连接2个吉他

Guitar Rig 2的硬件做工很好，铝制外壳很漂亮。它的USB音频接口稳定性也很好，在最小延迟下可以达到低于3ms的ASIO延迟时间。

当我要对Guitar Rig 2硬件上带的声卡进行RMAA回路测试，看看它的音质到底如何的时候，我发现无法测试。因为Guitar Rig为了保证硬件输入声音适合软件处理，对声音进行了低切过滤。

Guitar Rig软件：

点这里下载我用Guitar Rig 2做的一段声音。

再往后，就是我们大家都很熟悉的Guitar Rig了。它完全是真正自由的吉他效果器，所有东西统统模块化，包括箱头，箱体与单块等等。如果你愿意，甚至可以在箱头和箱体之间加个混响！（当然恐怕不会有人这么干……）。Guitar Rig 2其实与1看起来差不多。由于1已经实现了模块化。2感觉只是在1的基础上加入了一些新的模块，值得一提的是Guitar Rig的踏板，基本可以认为是kore的实验版。可以实现声卡，控制器等等一些功能。而且踏板可以一定程度上增加信噪比，使Guitar Rig 2 变成一个完美的吉他处理系统。（AmpliTube的踏板功能似乎比Guitar Rig2强一些，可惜还没出呢……唉）



Guitar Rig一共有6类模块，分别是Amps（箱体），Distort（失真），Mod（周边），Filter（均衡），Volume（音量相关），Other（其他，延迟和混响等等） 还有低频震荡器与包络控制。

Guitar Rig有一个虚拟机架的概念，各种模块都可以随意地进行组合，串联，并联，把两条串联通路suming到一起，等等等等都可以用它里面的“分割”模块来自由组合。在Guitar Rig2里面更是加入了以高低频率来组合两条通路的设计，真的是太完美了！这个设计比amp2的几个走向选择要好的多。



Guitar Rig中的失真模块之一，可调性很大。

另外，Guitar Rig的所有模块的基础参数都可以被低频震荡器和adsr调制。比如，我们没有硬件来控制哇音踏板的哇音，我们可以使用一个低频震荡来调制哇音。用不着手动调整啦！



其实我最喜欢，也是很多国外论坛都比较赞同的是，Guitar Rig的箱体，mic模拟是全世界最好的。听说有个录音学校就用Guitar Rig的mic模拟来讲拾音。可能大家觉得rig的mic和箱体模拟非常不爽，其实，真mic录音出来的效果就非常类似于rig的声音。那么为什么cd里面的吉他声音这么爽呢？其实是因为现在的吉他拾音一般都是多mic，甚至多箱体拾音!

Guitar Rig的设计正是满足了多mic摆位狂人的“瘾”。





很多真实拾音中的mic大招都可以直接应用到Guitar Rig里面来。但是相对的，Guitar Rig丧失了一个重要特性——简易性。 AmpliTube2一分钟调出来的音色，Guitar Rig可能要调上15分钟……因为所有的音色全部是从最基本的模块组合起来的，录制一个合格的失真节奏音色可能需要4个mic从不同角度与距离来拾音……



实际上NI也把它的箱体模拟技术应用在了它的其他软件上，不信？看看你的b4合成器的箱体模拟，呵呵。

Guitar Rig的箱体模拟虽然很好，但是箱头模拟却感觉非常一般……声音不如AmpliTube2那么有质感。



这样，在介绍了Guitar Rig 2和AmpliTube2之后，我们就能感觉到Guitar Rig 2与AmpliTube2在设计上的一些区别了：

简单对比：

	Guitar Rig 2	AmpliTube 2
背景		
制造厂商	Native Instruments	IK Multimedia
发布时间	2005年10月10日	2005年1月23日
上市时间	2005年12月21日	2006年3月31日
媒体报价	499欧元（含硬件控制器）	399美元（不含硬件控制器）
国内代理	千帆乐器 4799人民币	无
运行方式	独立运行，插件	独立运行，插件
插件格式	VST DX RTAS AU	VST RTAS AU
支持系统	Windows XP Mac OS X（不支持Intel-Mac）	Windows XP Mac OS X（不支持Intel-Mac）
功能		
音箱模拟	21种	16种
话放模拟	8种	21种
话筒模拟	9种	6种
常用效果	35种	32种
调音器	有	有
硬件控制器		
是否上市	是	否
随软件赠送	是	否
连接方式	USB 2.0	USB 2.0
切换踏板	6个	10个
其它控制器	2个	6个
音频接口精度	24bit/96kHz	未知
模拟信号输入接口	吉他输入	2个带放大的线路输入
模拟信号输出接口	2个高电平输出 2个低电平输出 1个直接输出	2个可选高低电平的输出
MIDI接口	是	是
耳机输出	是	是
数字输出	否	是
大屏幕显示	否	是
旋钮控制器	否	是

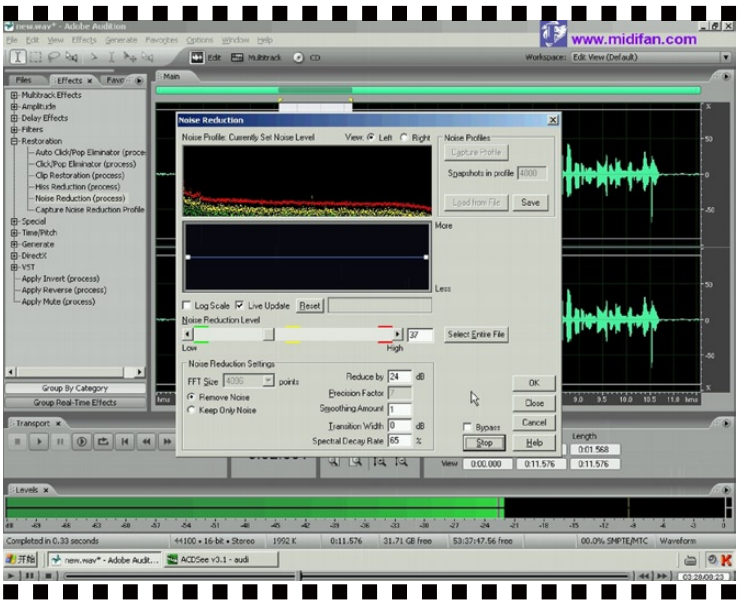
如果仅仅从功能上看，Guitar Rig 2基本是领先于AmpliTube 2的，但是换个角度，AmpliTube2比Guitar Rig 2胜在了易用性上！易用性不一定就是贬义词，AmpliTube2可以让人在最短的时间里作出最棒的效果。我第一次使用amp2就调出了与其演示曲很类似的声音，而我至今也没在Guitar Rig中调出NI官方演示曲的效果……

对比的结果终于出来了，Guitar Rig 2与AmpliTube 2虽然同属模块化吉他效果器，但很明显Guitar Rig的模块更加自由多变。AmpliTube2虽然也是模块化，但依然逃不出“效果器——音箱”的路由。而且，Guitar Rig的模块每样都可以添加数个。从功能上来说Guitar Rig依然远超AmpliTube 2。但是功能多并不意味着绝对的优势。两者的设计理念不同。AmpliTube更加像一台传统的合成效果器，声音也更加塌实、实在，“胶”味更足，更加会使人产生舒服的印象。而Guitar Rig则突破了传统效果器的限制。更加像是从数字的角度模拟了整个吉他录音系统，包括信号路由，单块连接，箱头模拟，箱体模拟，mic模拟等等吉他录音中必须使用的东西，从而制作出吉他音色。所以假若两者如果使用同样的效果模块，AmpliTube2明显会更加占有优势，而Guitar Rig则显得声音发散。但是精细调节起来，Guitar Rig的声音明显比AmpliTube2增加质感。举个例子，Guitar Rig就好比BFD，非常非常真，但是如果不调整就拿来用，是根本不可能好听的。Amp就好比一些本来效果就好，我们只需要拿来用就可以的音源。两者各有千秋。根据个人需要，我建议希望立即得到好效果的用户使用AmpliTube2，而熟悉录制电吉他以及希望得到精细调节功能的用户使用Guitar Rig2。

Adobe Audition 2.0 视频面对面（十一）

优化声音波形

http://forum.midifan.com/magazine/video/11/11.html



老版本的阅读器可能不支持链接，请手动输入地址到浏览器，观看视频：

作者：Pediatrician

格式：swf

时间：8分23秒

规格：1024X768

大小：6.26M

简介：

详细讲解如何对硬件条件不满意的情况下录制的声音进行优化，包括调整DC偏移，降噪，0交叉的意义与应用，意外爆破音的处理。

Adobe Audition 2.0 视频面对面（十二）

增加声音特效

老版本的阅读器可能不支持链接，请手动输入地址到浏览器，观看视频：

作者：Pediatrician

格式：swf

时间：8分49秒

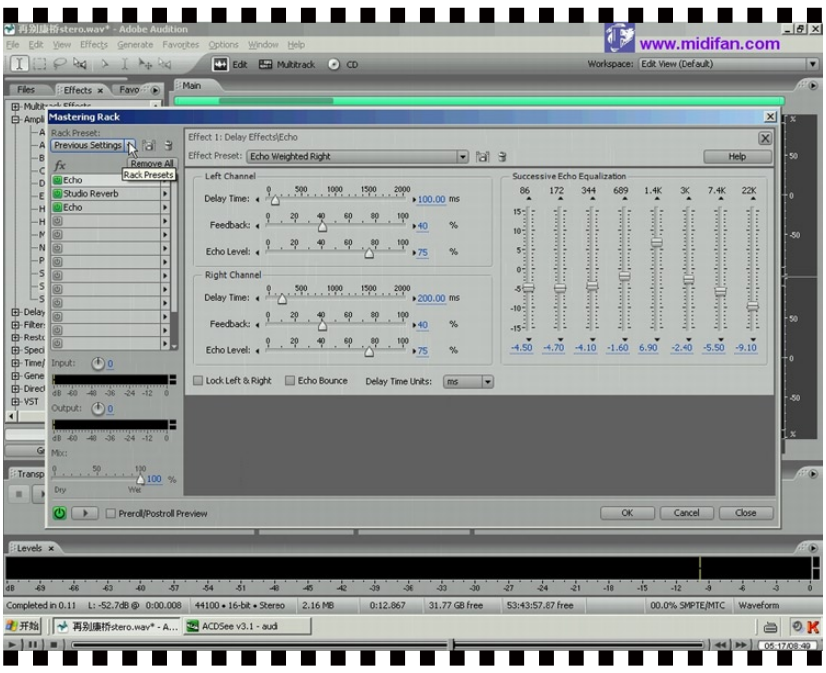
规格：1024X768

大小：4.27M

简介：

详细讲解编辑模式下如何增加声音特效和声音特效的基本设置。

http://forum.midifan.com/magazine/video/12/12.html





水果n连音神功

由于水果没有比较方便的做多连音的工具，所以我们在这里给大家讲解一下可以做出n连音的方法，以及提高多连音制作效率的秘诀——n连音神功。

Wode/文

制作n连音需要了解的一些基础知识：

默认情况下，你在钢琴卷轴中单击鼠标左键可以输入一个标准的四分音符，如图所示：

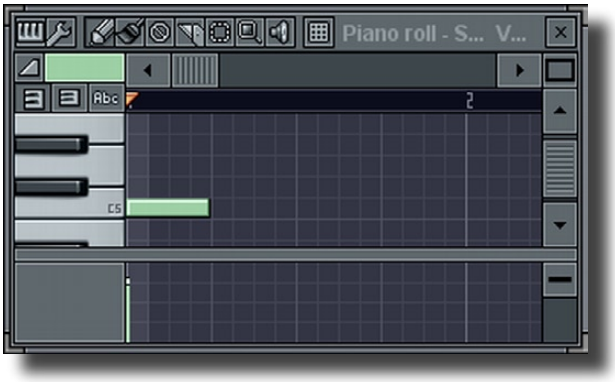


图01：钢琴卷轴中输入的标准的四分音符

如果不是，我们可以通过将鼠标移动到音符的左边，当鼠标变成 ✥ 形状的时候左右拖动来改变音符的长度。或者当鼠标变成形状，左右拖动音符来改变音符的起始时间。但是，这样是不很精确的，这样我们就会用到音符属性窗口，通过双击你要修改的音符来打开，如图：



图02：音符属性窗口

在Duration（持续时间）的数值显示器上单击鼠标右键会出现一个Reset（重置）选项，如图：

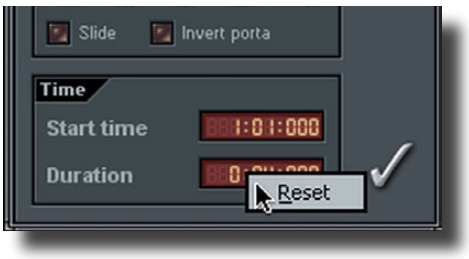


图03：音符属性窗口Reset选项

我们选择这个选项之后，音符就会被调节到默认的标准的四分音符长度了。其中数值显示器上从左至右依次为“小节：1/4个四分音符：最小音分”，如：0:03:023 。

我们谈以上类容的目的是为了你便于理解后面我们要讲解的内容。

刚才谈到的最小音分的值是根据设置窗口中的Timebase选项决定的，可以通过OPTIONS（选项）菜单中的Project general settings（工程常规设置）选项来打开并进行设置。如图：



图04：OPTIONS菜单-Project general settings选项

默认情况下Timebase选项设置的是96。你可以通过下拉菜单来更改它的数值，如图：

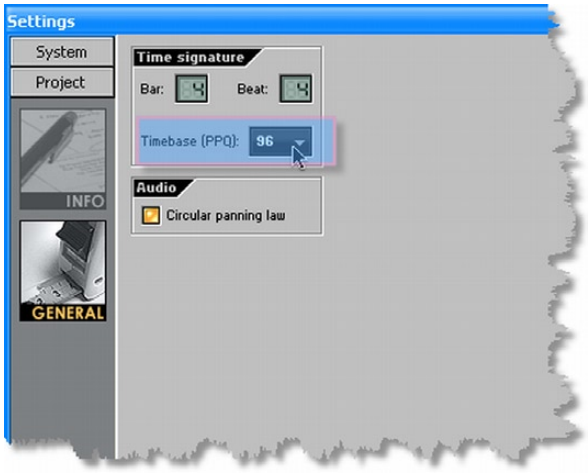


图05：设置窗口中的Timebase选项

这个数值意思是说将一个四分音符分为多少份，所以：最小音分=1/ Timebase。由于1/4个四分音符×4=四分音符=Timebase，所以1/4个四分音符=1/4Timebase 即当Timebase=96的时候，1/4个四分音符=24。其他Timebase数值类推。

别被上面的几个算术题给弄晕了，其实并不复杂，应该是很好理解的。说的这么详细是为了接下来给大家讲解怎么在水果中做出n连音的，比如说三连音、五连音等。

制作n连音的方法：

大家了解到了Timebase值，那么我们来看看怎么做n连音。

比如我们现在要将一个四分音符的长度分为3份，即输入一个四分三连音。如果Timebase=96 ，那么就是说将一个四分音符分成了96份，要将它分为3份，很显然就是96÷3=32，也就是说你输入3个持续时间为32最小音分的音符并且将它们排列起来就可以得到一个三连音。

比如我们先在第一小节输入一个音符并且双击它打开音符属性窗口。并且将Start time（开始时间）设置为“1：01：000”，Duration（持续时间）设置为“0：01：008”，如图：



图06：音符属性窗口中设置第一个音符

将持续时间设置为“0：01：008”的原因是，最小音分满24会折合为1个1/4个四分音符，而24+8=32。

接下来我们将开始时间设置到“1：02：008”，因为前一个音符结束的时间是在“1：02：008”，所以这个音符要从“1：02：008”处开始。和前面那个音符一样我们将它的持续时间设置为“0：01：008”。如图：



图07：音符属性窗口中设置第二个音符

用同样的方法输入第三个音符，如图：



图08：音符属性窗口中设置第三个音符

大家可以看到，这样我们就得到了一个标准的三连音了。用同样的方法可以输入其他三连音，如图：

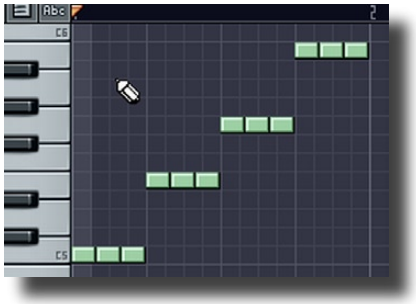


图09：输入好的三连音

我想大家应该已经领会到了怎么来输入三连音了，接下来我们来输入一个五连音。假如我们要将1个四分音符分为5份，也就是输入一个四分五连音。怎么来进行呢？我们看96÷5=19.2，已经不是整数了，这样不行。遇到这样的情况，我们可以通过更改Timebase的值来解决，比如我们将它设置为120，现在120÷5=24，就得到的是整数。所以只要将每个音符的持续时间设置成“0：00：024”，并且将他们排列起来就可以得到一个标准的四分五连音。如图：



图10：设置五连音

具体细节可以仿照上面三连音的输入。

但是有人要问了，唉~~我要输入一个四分七连音又该怎样来输入呢？Timebase的值无论设置的是哪个都除不尽有小数部分怎么来输入啊？

遇到这样的情况，我们采取的办法是将Timebase的值设置为768，也就是其中最大的一个值。为什么要设置成768呢？是这样的，（768+2）÷7 = 110或者（768-5）÷7 = 109，这样就能够除尽并且不带小数了。数学问题这个大家都知道。这样算出来的数值，偏差值是比较小的。我想768分之1应该是很难听出来的，我们用的也就是这样原理来进行输入。

下面我们就以110或者109的最小音分来输入这个七连音。输入的方法也基本和前面相同。当然我们在输入的时候也可以采取一些其他的手段，比如你减掉5个最小音分后你可以将这减掉的5个最小音分加到其中5个音符中每个音符加1最小音分，这样基本上可以说是听不出来破绽的。下面是输入的七连音，如图：

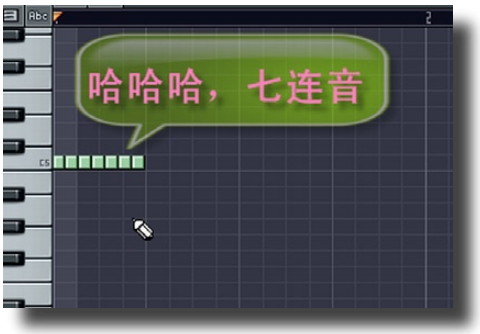


图11：偶输入的七连音

其他的多连音都可以按照上面的方法来输入，但是可能输入起来比较烦琐。有没有更简单的方法或者是提高效率的方法呢？

其实你只要输入了一个比如说七连音，如果后面还要用这个七连音，那么你就可以复制、粘贴了。

为了输入更加方便，我们特别制作了水果n连音包(点此下载)。大家一旦拥有了就相当于练成了n连音神功，效率+++倍。可以让你方便的输入二、三、四、五、六、七、九、十、11、12、13、14、15连音，虽然提供的连音有限，不过一般情况下不会用到再比这么复杂的连音了。如图：

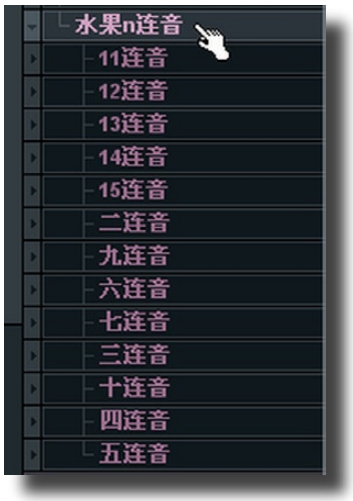


图12：水果n连音包

使用水果n连音包-提高输入效率-练就n连音神功：

水果n连音包是费了很大力气精心制作的，可以让你输入连音的速度提高很多，如果再结合水果的快捷键会让你的输入速度更加快。

下面我们给大家介绍一下它的用法。下载水果n连音包后请将其解压缩到水果的Data\Patches目录，或者Data\Patches\Scores目录。然后我们打开水果的时候就可以在Browser（浏览器）中或者是Browser的Scores段中找到水果n连音段了。大家也可以根据自己的习惯或者爱好解压缩到相应的目录。如图：



图13：解压缩目录与Browser中对应位置



图14：解压缩目录与Browser中对应位置

现在我们就可以开始使用了。在Browser中展开后，大家可以很容易的找到自己需要的连音。每个连音段中都会有名为1、2、4、8、16、32、64、128的八个文件，它们分别代表，全音符、二分音符、四分音符、八分音符、十六分音符、三十二分音符、六十四分音符、一百二十八分音符。使用方法就是直接将其拖动到钢琴卷轴或者步骤音序上就可以了，如图：



图15：Browser和Import MIDI data

当拖动文件到钢琴卷轴后会弹出上图中的Import MIDI data窗口。特别要注意的是其中的Blend with existing data（与已有数据混合），最好选中此项，以免将你输入的其他音符被替换了。如果不小心没有钩选此选项而覆盖了你输入的其

他音符，那么也不用怕，你可以按快捷键（Ctrl + Z）或者选择EDIT（编辑）菜单中的Undo（撤消）选项来撤消。

你需要输入连音的时候也许是在乐曲的中途，请记住一定钩选Blend with existing data，然后按 ☑ 确定。当连音插入的时候音符是在第一小节的，为了提高速度在你插入连音后请直接按快捷键（Ctrl + X）将刚刚插入的连音剪切到剪贴板，然后在你要插入连音的小节按快捷键（Ctrl + V）粘贴。如果要输入的连音不是从你要输入的小节开始处开始的，那么你只要再用鼠标左右拖动来调整位置即可。

补充说明一下。2连音和4连音是将原来的3份分为2份和4份；3、5、7连音将原来的4份分为3、5、7份；9、10、11、12、13、14、15连音是将原来的8份分为9、10、11、12、13、14、15份。使用的时候请根据自己的需要。如果你要输入更为复杂的连音，那么之后按照前面讲解的基本方法来做。

好了~也许你累了，不过不要紧。练功是有好处的，到此你就已经练成了n连音神功！F



点这里进入论坛讨论此文

用外部工具直接快速导出分轨

musiXboy/文



我敢说所有Cubase SX的使用者都用的上今天我们要讲的小技巧，这可以节省大量的时间和体力。

Cubase SX几乎可以说是世界上最好用的编曲软件，但是……有一件很重要的事情它做不了，直到Cubase SX升级到3.1版本也无法实现，那就是直接导出分轨。

在Cubase SX导出分轨确实是一件头疼的事情，尤其在你做了数十条VSTi和音频音轨的时候，你要一轨一轨的单独导出分轨，又耗费时间又耗费精力。

现在，我们有了一个外部工具，利用它就可以快速导出分轨。黑毛老师曾经发明过一个利用冻结功能导出VSTi分轨的小窍门，但是我们今天要说的这个技巧，不仅可以导出VSTi分轨，还有音频的分轨。

下面就跟我来吧。别看整个操作需要十多步，其实每一步都很简单，在你熟练之后导Cubase SX的分轨将是一件非常开心的事情，Cubase SX将像一个机器人那样听话的导分轨。

第一步：打开Cubase SX，并打开你的工程文件。确认Cubase SX Project窗口并没有最大化，如下图。



Project窗口并没有最大化，正确



Project窗口最大化了，错误

第二步：另存为一个新的文件（为了防止意外，做个备份还是很有必要的）。

第三步：选中那些不需要导出分轨的音轨（包括音频轨、VSTi轨、MIDI轨、效果轨、folder轨、标记轨、分组轨）挪到最后，如下图，只留下需要导出的MIDI和音频轨，其它轨都靠后站。



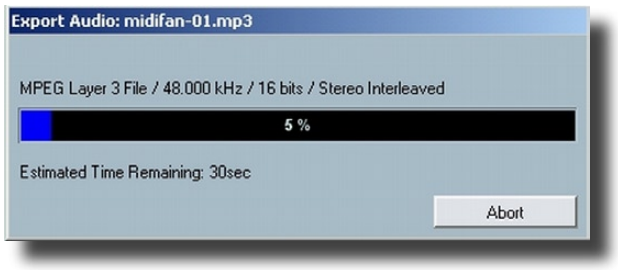
只留下需要导出的MIDI和音频轨，其它轨都靠后站

第四步：像以前导出整个混音那样去设置你的工程文件，比如调整好左右标尺的位置，以便让Cubase SX知道你要导出哪段时间里的声音。

第五步：点击File——Export——

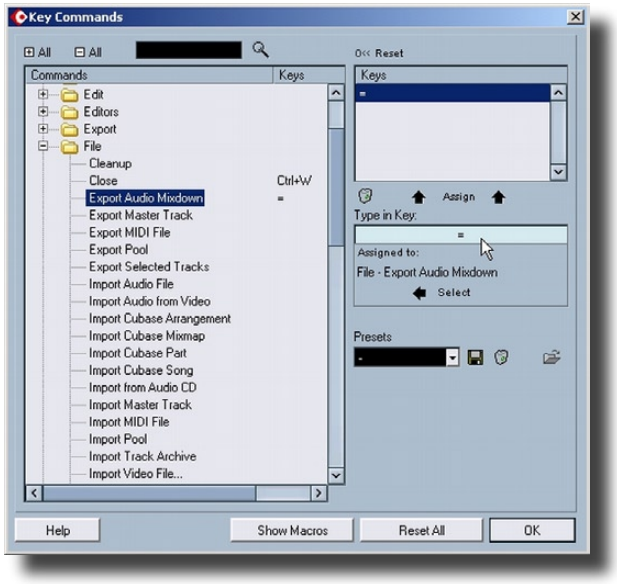
Audio Mixdown，按照常规导出整个音频混音。这么做的目的是为了使Cubase SX记住你的导出设置，是mp3还是wav？是立体声还是单声道？等等。

第六步：设置完毕后，导出文件。找个小纸条记录下导出时对话框中最初显示的导出时间（注意不是音轨的时间长度，而是导出需要的时间长度），如下图。



记录下Estimated Time Remaining最初的值

第七步：点击File——Key Command，在窗口里找到File目录的Export Audio Mixdown项目，选中它，在右边Type in Key中点一下鼠标，按下键盘上的“=”（等号）键，再点Assign按钮，最后点OK确认。这是设置“=”为音频导出的快捷键。



设置“=”为音频导出的快捷键

第八步：下载SX Unattended Audio Export Tool（SX无人职守音频导出工具）这个软件，地址：http://forum.midifan.com/magazine/down/06/SX_Unattended_Audio_Export_Tool_0.6.zip

第九步：选中Cubase SX中的第一个音轨（再次确认所有排在前面的音轨都是你需要导出分轨的，而且都没有被静音和独奏），最小化Cubase SX，并双击打开SX Unattended Audio Export Tool软件，它将检查一下Cubase SX是否正常工作。



双击打开SX Unattended Audio Export Tool（SX无人职守音频导出工具）软件



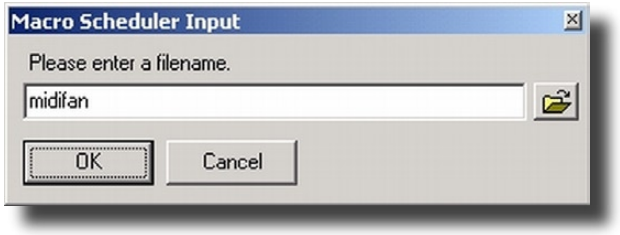
软件提示你没注册，不用管它

第十步：一切顺利的话，它会问你：你要导出多少条音轨？如实回答之后点OK按钮，如下图。



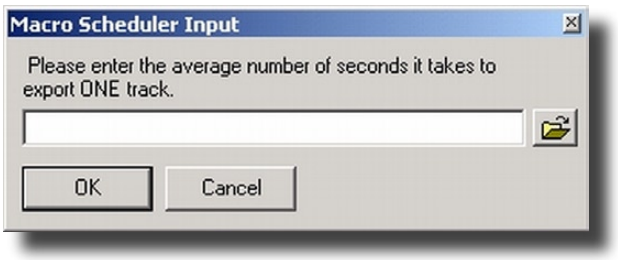
你要导出多少条音轨？

第十一步：它继续问你：你要导出的文件名，继续如实回答，点OK按钮，如下图。



你要导出的文件名？

第十二步：它接着问你：平均导出一条音轨要用几秒？拿出在第六步时记录的小纸条，填写下比记录时间更长一些的时间。比如你之前导出用了5分钟，也就是300秒，那么这里最好写330秒，也就是5分半钟。这个时间决定了导出第一条音轨后等待多长时间再导出第二条音轨，所以最好打出一些富裕。



平均导出一条音轨要用几秒？

第十三步：最后通牒，你真的要导出分轨吗？当然点Yes。



你真的要导出分轨吗？

第十四步：软件开始工作了，Cubase SX自动最大化出来，按照你在第十步设置的音轨数，自动一轨一轨设置为独奏并导出，整个过程Cubase SX好象是一个机器人，而你现在可以去泡杯茶休息一下。

第十五步：全部完成后，软件提示你全部工作完成了！



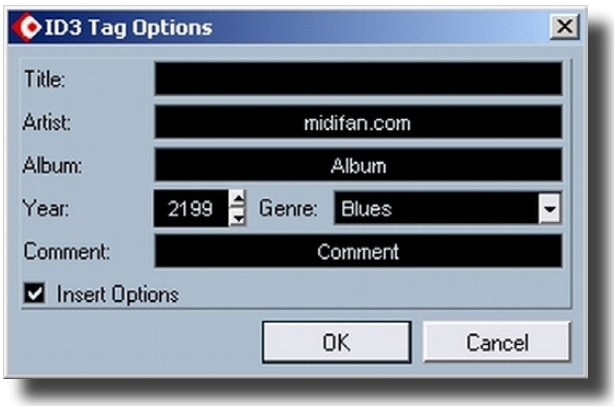
大功告成

第十六步：到你在第五步导出音轨的目录里，就可以找到所有分轨了。它会自动按照你在第十一步设置的名字+音轨的数字编号进行命名，如下图。



最后导出的分轨

可能附加的一步：如果你导出的不是wav而是mp3或wma，那么“无人职守音频导出工具”恐怕就要改成“有人职守音频导出工具”了。因为在导出mp3或wma的时候，Cubase SX还会额外让你插入作者信息，在你看到下图画面的时候，必须快速按下OK键，否则就无法成功导出分轨了。



导出mp3或wma，到这里必须手动按OK键

后记：

最后声明一下，这个外部工具是老外MiLO利用Macro Scheduler软件写的一个脚本，向他表示感谢。可惜它还无法工作在Nuendo中，也无法工作在Mac OS X系统中，非常遗憾，万事都无法完美。

作者还在不断升级，也许以后功能会更完善。有任何新的消息，我们midifan.com会第一时间告诉大家。

历史车轮：Native Instruments发展历史回顾

Native Instruments标志

（根据Native Instruments官方资料由musiXboy编译整理）

Native Instruments的故事应该由上个世纪的1995年在德国柏林的一个很小的住宅式办公室开始讲起。Stephan Schmitt，一位开发调音台的硬件工程师，同时也是一名经常做现场演出的音乐人、合成器玩家，发现当时的乐器有很多限制，那些电子乐器不仅体积庞大，价格昂贵，而且大大限制了个人在演出时的音乐灵感，这使Stephan Schmitt感到非常的不爽。

Stephan Schmitt在十几岁的时候，就自己独立开发出了第一台电子发声设备：一台模块化的合成器，他非常喜欢摇滚、爵士、实验电子音乐。在他上大学的时候就是柏林爵士乐团的积极份子，负责演奏各式各样的合成器。在Stephan Schmitt获得电子工程师的学士学位之后，他作为硬件工程师进入了一家生产High-End大型模拟数控调音台的公司。



Stephan Schmitt

正值此时，Stephan Schmitt意识到个人电脑可以作为一个强大的电子乐器使用，性能逐渐提升的电脑已经可以实时进行声音合成的工作，而且完全不受任何限制。Stephan Schmitt马上找到了程序员Volker Hinz，和他在业余时间一同研究他们的模块化合成器软件，当时叫做GENERATOR（后来改名Reaktor），GENERATOR允许人们自由的创建并组合各种声音合成模块，通过灵活的操作建立起自己需要的合成方式。



GENERATOR最初的界面

1996年，GENERATOR已经可以在当时奔腾芯片的电脑中完美的运行，Stephan Schmitt于是将它带到了德国的MusikMesse展会上，GENERATOR一经展出马上吸引了大批的音乐人和记者。与此同时，Stephan Schmitt与Volker Hinz成立了Native Instruments有限公司，并将GENERATOR作为一个商业软件于1997年夏天正式投放市场，大批的“粉丝”开始讨论如何用GENERATOR制作电子音乐和techno，Nine Inch Nails（九寸钉）乐队的Trent Reznor也是其中的一员积极份子。



Native Instruments核心成员合影

更多的人看到了GENERATOR蕴藏的巨大潜力，有4人在1997年先后加入了Native Instruments公司，大大加强了技术实力，其中Daniel Haver任常务董事。同年Native Instruments转变为股份有限公司，由6部分股权组成，总部定于德国柏林Mitte。到1999年，Native Instruments的软件已经分销到世界20多个国家，雇员达到15名。

2000年，Native Instruments将目光投向了一些著名的古老合成器，根据B3拉杆风琴和Prophet 5模拟合成器，开发出了B4拉杆风琴虚拟乐器和PRO-5模拟合成器虚拟乐器，公司的雇员比去年翻了一翻还多，达到40多名。



Berlin's Business Plan

Start-up companies are the key. BY STEFAN THEIL

IT'S SATURDAY NIGHT IN Berlin. The narrow streets of the ultrahip Mitte district are packed with partyers flocking from bar to bar, past the trendy galleries, club-wear boutiques and all-night Turkish eateries. In a tenement apartment above a noisy café, Stephan Schmitt and a handful of musicians and computer hackers crowd around a keyboard and a couple of PCs, creating a strange cocktail of electronic bleeps, twangs and swooshes.

A techno jam session? Not exactly. These young people are making something Berlin needs even more than new dance tracks for its nightclubs: jobs. Schmitt is founder and technical director of Native Instruments, a 14-person software start-up that's developed a software-based sound synthesizer. He and his team are racing to finish the code for their newest release, to go on sale in 20 countries in a month. Music magazines have showered the company's products with awards, and Schmitt says he's been approached by both Microsoft and Apple for possible licensing deals. "Our goal is to become the worldwide standard," he says with a grin.

Berlin is rooting for him. The local economy shrank by .4 percent last year, after several years of zero growth—the worst economic performance of any German region. After Bonn cut off the subsidies that propped up both sides of the city before reunification, much of the old-line manufacturing that was the mainstay of the local economy collapsed. Facilities like Siemens' machine-tools factory simply shut down; several local textile makers shifted production to low-cost Poland. All in all, the city has lost some 300,000 manufacturing jobs in a decade. The 50,000 new government jobs now coming to Berlin are no consolation as the city slashes its own bloated public sector.

Hopes of drawing a new corporate elite to the city that was once Germany's economic powerhouse proved unrealistic. After moving away in the 1950s—its plants were in ruins, and the Soviet Army controlled all transport routes through East Germany—electronics giant Siemens remains firmly entrenched in Munich. Ditto for Deutsche Bank, now in Frankfurt. Sony will open a gleaming new, $800 million European headquarters complex on Berlin's Potsdamer Platz next year, but it's a rare example of a global player setting up shop in Berlin. Klaus Kräge, head of Jones Lang Wootton's Berlin office, says that prices for commercial real estate have fallen to half their early-1990s peak. Instead of becoming the grand hub of East-West commerce many envisioned, Berlin often feels cut off from the rest of the world. Today the only scheduled intercontinental flights to the city are from Havana and Ulan Bator.

Luckily, Native Instruments isn't the only new company trying to turn things around. "Whether it's in information technology or biomedicine, there's been a boom in new high-tech companies since the wall came down," says Birgit Schultz, analyst for the Prognos consulting group. Over the last 10 years, some 100 biotech firms alone have set up shop, clustered around the city's universities and research clinics. One is Mologen AG, currently operating out of a lab at Berlin Free University's molecular-biology department. There, Prof. Burghardt Wittig and about three dozen recent graduates are working 16-hour days, developing genetic vaccination techniques. To raise money for research—and for the new lab building going up next door—Mologen last year became Germany's first university biotech spin-off to go public, listing on Berlin's over-the-counter stock market. "We all own stock and work like idiots," Wittig says. Once the new building is finished, he expects to raise more capital—and add new staff.

The authorities like that so much that they're trying to help the city's entrepreneurs. Wittig says he was shocked when it took him only two months to get a building permit for the new lab. The city has built 21 "start-up-centers" all over town that offer cheap quarters and basic office infrastructure for hundreds of fledgling companies. New, private venture-capital firms now scour the city for deals, their investments often leveraged by generous public-sector matching funds.

Of course, it takes more than money and real estate to create new companies. It takes ambitious, creative people—just the kind who want to move to a place like Berlin. "Look outside," says Native Instruments' Schmitt, pointing at the crowded sidewalk café below. "This is a place where it's fun to work and live and go out." And—if Berlin is lucky—maybe even get rich, too.

Schmitt (left) is creating sorely needed jobs

1999年纽约时报对Native Instruments进行报道采访

在此之后的2002年，Native Instruments并没有满足于在合成器软件市场的成功，他们的触角延伸到了DJ领域、采样器领域、鼓机领域，分别推出了Traktor DJ、Kontakt、Battery等软件。

2004年，Native Instruments又掀起了软件吉他效果器的革命，推出了模块化的吉他效果器软件Guitar Rig。Guitar Rig不仅在电脑音乐圈里被大家所了解，在吉他圈里更是大有吉他手革命的架势，甚至连一些主流媒体的报纸、广播都介绍了Guitar Rig为吉他手带来的重大革命。

2005年，Guitar Rig 2的出现，不仅增强了软件的功能，更是创新的推出了配合Guitar Rig 2使用的硬件音频接口+踏板控制器，Native Instruments摆脱了软件制造商的帽子，开始进入硬件市场。

Stephan Schmitt虽然目前是Native Instruments的总经理，但他更愿意全身心投入到产品开发中，Reaktor、Guitar Rig或是最新的Kore都是由Stephan Schmitt亲自主持开发的。



总经理Stephan Schmit(左)和CEO Daniel Haver(右)

2006年，Native Instruments迎来了公司成立10周年的大好日子，相信Native Instruments会为我们带来更创新、更前卫、更方便的音乐产品。

超级人声处理机 Sonar V-Vocal(二)

涂立康/文

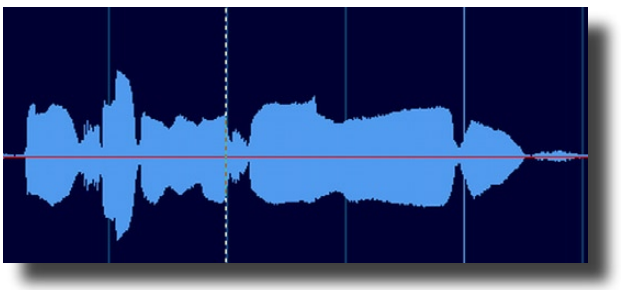
在录制人声的过程中您是否迫切的想解决一些看似天方夜谭的想法？是不是对一些歌手的音色不满意？男声太奶油，一点阳刚之气都没有，女声又不够温柔。一些歌手总是跑调，而一些歌手的气总是短了那么一截。当无法改变歌手水平的时候，那么我们只能改变自己了。如果您使用Sonar5来录制人声，上述问题的解决就变得是游刃有余了。

在上一期中我们较为详细的介绍了V-Vocal的功能及使用方法。本期主要就一些实际问题的解决之道展开讨论。

下面我们就来看看在实际的录音中常常会遇到什么样的问题？怎样利用V-Vocal来解决这些问题！

问：我们创作了一首歌曲，找了一个男歌手来演唱，演唱的不错，情感也很到位，但是歌手的声线不符合该曲的曲风。我们需要的是一个非常成熟的男性声音，但是歌手的音色偏亮偏薄。尽管用均衡器增加了中低频部分的音量，但是总体效果还是不够理想，请问有什么办法可以改变歌手的声线，使得声音听起来更厚实一些？

答：最根本的解决办法当然就是另外找一个声线符合要求的歌手重新演唱一次。不过由于成本、时间及其他客观原因不能实现的时候，我们可以利用V-Vocal的强大处理功能使得歌手的声线发生一定程度上的变化。在通常语言类的录音中，为了更加突出男播音员厚实的中低音区以及胸腔共鸣特点，我们往往会做一些技术处理。一个公开的秘密就是把录下来的音频进行降调处理，一般幅度在1～2个半音之内。这样播音员的声音听起来就会更加富有亲和力，低沉而有力。但是我们现在需要解决的是歌曲演唱的问题，受歌曲调式的限制，我们当然不会采用降调的方式来处理了。利用V-Vocal的共振峰调整技术我们就可以在不改变音高的情况下，使得歌手的声线变得更加低沉。



打开V-Vocal的共振峰模式，如上图显示的那样，在中间有一条红色的线，一边播放一边调低这条红色的线，我们就可以听到歌手演唱的音色逐渐变得低沉起来。当然不能调的太过分，否则听起来就像磁带机快没有电的效果。另外还有针对女声的调节也可以用这个办法，假如需要把一个女声的声线向小女生的嗓音靠，那么我们可以把这条红线往上移动。不过往上移动后产生的失真也比较大，因此调整的幅度尽量不要太大，适可而止。其实这种办法可以适合大多数的女声录音，除非歌曲的曲风像西北风那样要求粗犷的女声，一般情况下，针对成年女性的嗓音都可以做如此调整，特别是在歌手又是顾客的情况下，略略把红线往上提一下，可以使得歌手的嗓音听起来更加年轻一些；并且歌手本身的嗓音特征并不会改变太多。这就好比数码照相一样，把被拍摄者的皮肤进

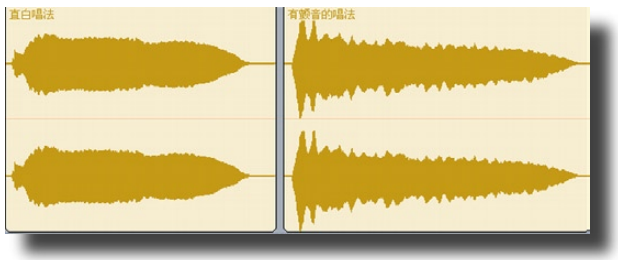
行光洁处理，减少皱纹，看起来更加年轻一些，女性顾客非常喜欢。在音频录音里我们如上述所说的办法进行处理，相信一定也会受到客户的欢迎。到时赚了大钱，可别忘了分一点专利费给笔者哈。

问：歌手总是跑调，有没有什么办法纠正？

答：给歌手纠正音高的软件有很多，常用的有Auto Tune、Melodyne。如果您恰好用Sonar5来录音，那么使用V-Vocal是非常不错的一个选择。毕竟Auto Tune最强大的功能在于自动纠调，当歌手演唱跑调跑得厉害的时候，就需要手动调节了。而Auto Tune手动画法准确但是麻烦，比较费功夫。Melodyne直到3.0的时候，才加入了自动纠调的功能，自动和手动功能结合的非常好，使用方便而且快捷。但毕竟是另外一个软件，和我们录音软件的结合不是很好，数据导入导出的浪费了我们不少时间。V-Vocal是Sonar5里面自带的软件，和母体软件结合的非常紧密，并且也同时具有自动纠调和手动调整的功能。详细的使用说明请参考上一期文章介绍的自动音高修正部分。这里我们讲解实际运用中所需要注意的步骤。如果我们的录音不是一气呵成的，那么首先我们需要把歌手演唱的音频部分合并成一个Clip（片段）。怎样合并这里简单说明一下，把所需要合并的多个音频Clip（片段）全部选中使其变成黑色，然后选取Edit（编辑）菜单下的Bounce to Clip（合并为剪贴）命令，稍等片刻就可以完成片段合并；然后我们要确定这首歌曲的调式调性，怎样确定请参考乐理方面的书籍，这里不作说明。如果整首歌曲是一个调，那么我们可以直接调用V-Vocal菜单下的Create V-Vocal Clip来创建一个V-Vocal片段。如果是歌曲有转调的，我们需要把歌手演唱的音频部分按照转调的方式来进行分割。每转一次就必须分割为一个片段，然后依次用V-Vocal来进行处理。在V-Vocal窗口里先用自动音高修正功能，在正确的调式下把声音自动纠正一次之后，播放试听，看是否有纠正错误的。如果发现有偏离正确音高的地方，我们选择Arrow Tools（选择模式），在按住Ctrl键的同时，用鼠标把音符拉到正确的音高位置上。

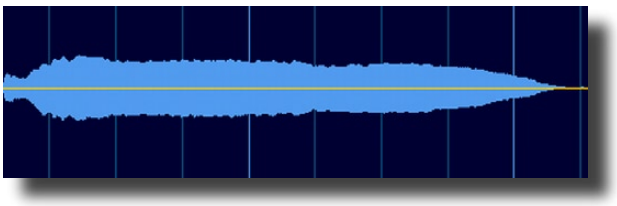
问：有的歌手在演唱长音时，唱得太直白了，请问有没有什么办法可以给歌声润色？

答：通过题干分析，这种情况往往是歌手在演唱长音时，没有添加颤音技巧造成的。利用V-Vocal我们可以给这个长音部分人为的添加颤音。那么首先我们要分析真正的颤音是怎样形成的。拿一个人声颤音的音频波形看一下，我们会发现，波形是整体上处于一种波浪状态。

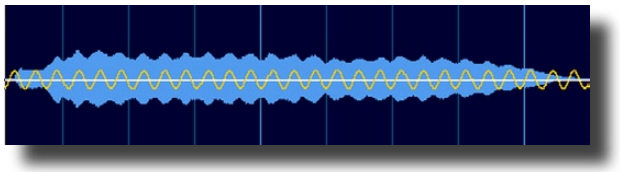


波形的纵向变化代表了振幅的变化，也就是音量大小的变化。也就是说，如果我们能给一个长音添加一个音量上的波动变化，那么是不是颤音就形成了呢？先做出来听一下！

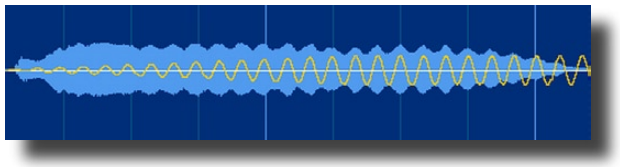
打开V-Vocal，找到需要被修改的长音部分。选择Dynamics模式，得到如图显示的样子：



选取LFO模式，添加一个波动曲线，幅度以听不到机械感为好。



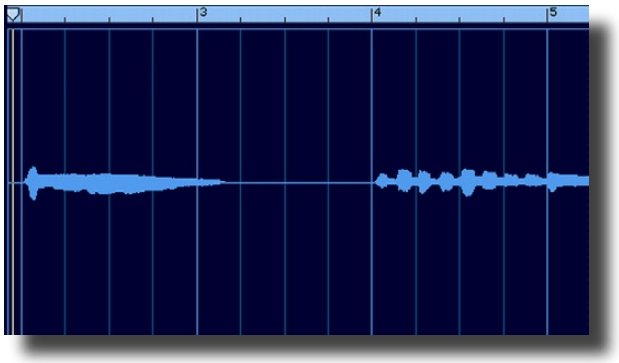
可以看到波形被LFO模式调制成带有波动形状的图示了。由于真正演唱的时候，颤音往往不是一开始就出现的，而是在发音之后逐渐形成的。为了模仿这种效果，我们需要使得添加的LFO调制的颤动大小是逐渐增大的。那么怎么操作呢，把鼠标移动到LFO调制形成的黄色正弦波的起始位置上，偏上或者偏下都可以，就是一定不能在中间位置，这时鼠标模式变成了形状。按住鼠标左键不放手，并把鼠标向右移动，就会形成LFO包络逐渐增大的效果。



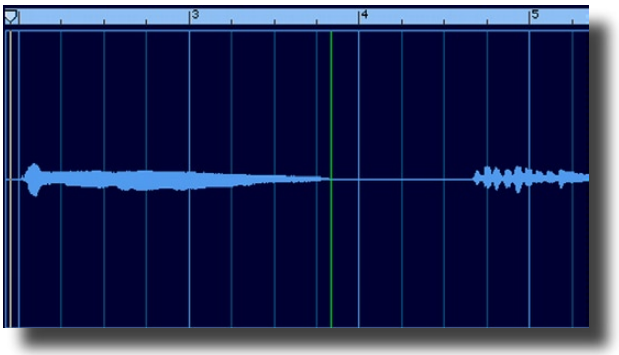
现在我们在音量上的颤音技巧已经全部做完了。试听一下，感觉和真实的颤音还是有较大的差别。这是怎么回事呢？原来，颤音的形成不光靠音量上的波动，还要靠频率上的波动才可以。也就是说，真正歌手演唱的颤音，其音高是不固定的，人耳听到这样持续的音高有规律的变化，自动会求出一个平均值，这个平均值就是中心频率，人耳通过这个中心频率高低来进行判断该音符是否准确。也就是说我们不仅仅需要在音量上做波动调制，还需要在音高方面做波动的调制。在音高修正模式下，使用LFO的方式按照前面讲述的音量包络方法，再同样的做一个渐强的音高波动包络就可以了。

问：有一些长音，歌手的气不够，不能够演唱到足够的长度，请问有没有什么办法可以延长歌手唱的声音？

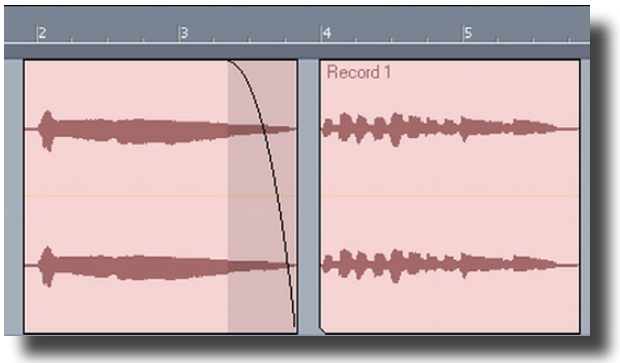
答：在Nuendo和Samplitude里面都可以很方便的延长歌手的声音。但是处理的精度不高，稍微延长的多一些，音质就下降的比较厉害，有些软件，比如Melodyne虽然在延长幅度很大的情况下都不会改变歌声的音质。但是有个致命的缺点就是，当演唱长音的时候，大多数的歌手都会加上颤音。一旦延长后，颤音的频率就下降了。当延长的幅度比较大的时候，这个频率就下降的非常厉害，给人非常不真实的感觉。在Sonar5里面我们利用V-Vocal可以较大幅度的延长歌手演唱的声音，通过人工颤音的方式重塑颤音的频率。这样既延长了声音，同时还可以保证颤音的频率不发生大幅度的下降。首先，我们把需要被延长的那部分声音音频从音频块里面切割出来，然后用鼠标向右拉长这个音频块，如果后面还有演唱的部分，也会被拉出来的，我们不需要理会这个，只要拉出来的音频块的长度为长音部分的两倍以上就可以了。点黑这个音频块（Clip）单击右键创建V-Vocal，选择Time模式。



由上图可以看出，长音到第三小节的时候就差不多结束了，中间还空了近一个小节。假如我们需要把这个长音延长到第三小节的第三拍才结束，该怎么做呢？在目前这个长音结束的位置上双击鼠标左键，就会得到一条绿色的分割线，然后移动这条分割线向右，直到长音结束的位置在第三小节第三拍的位置上。第四小节的演唱内容同样被移位并压缩了，不过我们不需要理会这个。然后播放试听看看长音中的颤音是否还是自然的。



如果颤音颤动得很缓慢，那么我们需要到音高修正模式下去重新安排音高颤动的频率。选中LFO模式 当鼠标移动到该音频上的中间线上下附近，鼠标会变成一个带铅笔一样的图标：。我们可以用这种模式来画出人工颤音，和前面讲过的一样，然后做一个渐强的LFO包络就算大功告成了。如果对尾音还有点不满意，我们可以选择Bounce to Clip（s）命令把刚才做的修改固定下来，把后面多余出来的字词删除掉，然后做一个淡出的包络。如图所示的那样，长音已经拖到了足够长的长度。



这样，我们的长音填补工作就算完成了。第四小节开始的后续声音都不会受到前面的改变而改变。

好，V-Vocal的实例讲解就到这里结束了，如果您在实际工作中还遇到其他方面的问题，可以直接和作者联系。QQ：357864123





Pro Tools小贴士：你也许不知道的实用快捷键



这一讲，我们来说一说Pro Tools的快捷键。也许你会说那些快捷键我早就在用了，我当然不会照着说明书一个一个告诉你这些快捷键，而是要告诉你一些说明书上没有写的、隐藏的很深的、却又非常有用的快捷键。

Mike Thornton/文 musiXboy/编译 (Sound On Sound杂志 2005年09月期)

Automation：

在使用插件，创建插件某个旋钮的automation自动变化曲线的时候，我相信所有人都是先点插件窗口上的Auto键，选择一个参数，加到列表里，再点OK，然后到音轨里去选择这个参数。实际上有一些非常简单的快捷键，只用两步就让你把automation参数显示到音轨里。



PC：Ctrl+Windows / Mac：Ctrl+Command

打开automation：

PC：Ctrl+Windows+Alt的同时在插件那个需要做automation的旋钮上点击

Mac：Ctrl+Option+Command的同时在插件那个需要做automation的旋钮上点击

使automation参数显示在音轨中：

PC：Ctrl+Windows的同时在插件那个需要做automation的旋钮上点击

Mac：Ctrl+Command的同时在插件那个需要做automation的旋钮上点击

要在automation曲线上加一个拐点呢？你一般都是选Grabber工具，那太麻烦了。

PC：右键单击automation曲线

Mac：Option+“/”

界面缩放与滚动：

编辑窗口太宽了，你要移动鼠标到下面的滚动条去左右拖动吗？有更好的方法，使整个界面向右滚动一个窗口的大小：

PC：Alt+Page Down

Mac：Option+Page Down

使整个界面向左滚动一个窗口的大小：

PC：Alt+Page Up

Mac：Option+Page Up

使单个音轨放大/缩小：

PC：无

Mac：Ctrl+向上/向下

使所有音轨放大/缩小：

PC：Windows+Alt+向上/向下

Mac：Ctrl+Option+向上/向下

在Mix窗口里使某个音轨出现在窗口的最左边（或者尽可能的靠左边）

PC：在Edit窗口中，Windows+点击某个音轨

Mac：在Edit窗口中，Ctrl+点击某个音轨

在Edit窗口里使某个音轨出现在窗口的最上边（或者尽可能的靠上边）

PC：在Mix窗口中，Windows+点击某个音轨

Mac：在Mix窗口中，Ctrl+点击某个音轨

其它非常实用的快捷键：

使Mix或Edit窗口中的电平表在宽模式和普通模式之间切换：

PC： Ctrl+Windows+Alt+点击某个音轨的电平表

Mac：Command+Alt+Ctrl+点击某个音轨的电平表

我们都知道如何复制出一个新区域到原始区域的后面，但复制到前面怎么做？

PC： Ctrl+Windows+Alt+点击某个区域

Mac：Ctrl+Alt+Command+点击某个区域



原始区域



向前复制

在对音轨进行重命名的窗口里，如何不关闭窗口继续对下一条音轨做重命名？

PC：Ctrl+向右

Mac：Command+向右

还有更快的，在对音轨进行重命名的窗口里，不需要按回车就可以对该条音轨的名字进行确认并自动跳到下一条音轨进行重命名：

PC：Ctrl+回车

Mac：Command+回车

新建音轨（这个大家估计都知道也常用）：

PC：Shift+Ctrl+N

Mac：Shift+Command+N

主要说的是这个，在新建音轨窗口中快速选择新音轨是单声道或立体声：

PC：Ctrl+向左/向右

Mac：Command+向左/向右

在新建音轨窗口中快速选择新音轨是音频、辅助、总线、MIDI：

PC：Ctrl+向上/向下

Mac：Command+向上/向下

在新建音轨窗口中快速选择新音轨的采样率和tick：

PC：Ctrl+Alt+向上/向下

Mac：Command+Option+向上/向下

Mac电脑趋势谈：

MacPro和MacOSX10.5Leopard

musiXboy/文

2006年8月7号，苹果在WWDC大会上，宣布了两件重头产品：Mac Pro 64位桌面电脑，以及64位内核的Mac OS X 10.5 Leopard（美洲豹）操作系统。

Mac Pro 64位桌面电脑：



Apple沿袭了MacBook Pro的叫法，将老的PowerMac G5系列的Intel芯片版本改称为Mac Pro。Mac Pro从外观上看，除了前面有两个DVD刻录机以外，其它地方几乎和老G5完全一样，毕竟G5的外观设计已经非常经典了，确实没必要再改。

而Mac Pro的内部却是大不同，按照传统PC的设计将电源移到了最上面，而且重新设计了可拉出的内存插槽，以及四个硬盘插槽，风扇则由原来的两个变成了一个（个头比以前的大了不少）。

Mac Pro啥配置？

- CPU：双核心Intel Xeon 2.66GHz x 2（4MB共享二级缓存）
- 显卡：NVIDIA GeForce 7300 GT 256MB
- 内存：1GB, DDR2 SDRAM 533MHz, ECC (2x512mb DIMMS)，最多32GB
- 硬盘：250GB 7200rpm SATA，另有三个硬盘位置，最多2TB
- 刻录机：Superdrive双层 DVD+/-RW驱动器
- 鼠标/键盘：Apple键盘和Apple Mighty鼠标
- 声卡/火线：搭配光纤输出的声效，FireWire 800/400
- 显示器：无
- 价格：2499美元

购买Mac Pro之后的第一件事情就是加内存，注意一定要使用Apple认证的内存，才能保证兼容性。另外我建议大家换无线的Apple Mighty鼠标。注意到Mac Pro依然配备了火线800的接口，这意味着Fireface 800还可以继续以最佳状态运行在Mac Pro上，而非之前人们猜测的那样——Apple并没有放弃800火线接口。

需要多少银子？

不加显示器的整机价格为2499美元，不得不说这个价格在如此的配置上是非常公道的。对比与Mac Pro类似的Dell Precision 490还要便宜，而且别忘了这个价格里还包含了Mac OS X操作系统。

Mac Pro有多快？

就拿音乐软件说吧，Mac Pro与以前的双CPU PowerMac G5 Quad相比，运行Logic Pro快了1.8倍，运行Soundtrack Pro快了1.6倍。

啥时候能买到？

目前Mac Pro已经在欧美上市了，由于中国市场需要进行3C认证，所以Mac Pro至少要在9月之后才能买的到。

Mac OS X 10.5 Leopard(美洲豹)操作系统：

这是所有苹果用户最关心的话题了，被Apple戏称为Windows Vista 2.0版的Mac OS X 10.5系统，新增了许多对音乐人来说很棒的功能。虽然我们在明年年初才能用上新的系统，但它展现出的魅力已经让我们流口水了。

64位系统：

首先它是一套64位内核的系统，当然也完全兼容现在咱们用的32位程序。但是我相信在10.5出现之后，最先支持64位内部精度的音乐软件会是Logic Pro，然后就是和Apple合作最紧的Digital Performer、Pro Tools，至于Cubase SX我不抱什么希望了，它得先解决支持Intel-Mac的问题，再来考虑64位的问题。

但是还别高兴太早，宿主64位化还不难，难点在插件的64位化，以及64位宿主对老32位插件的支持情况。64位的Sonar 5有一套桥接方式来支持老的32位插件，我想在苹果系统上实现类似的桥接技术并不难，在成千上万种插件向64位过渡之前，桥接技术将是非常关键的。

时间机器：

苹果的时间机器（Time Machine）远远超越了Windows XP的系统还原（这个无用的功能我想大家都不会去用的）。苹果的时间机器可以每天对每个目录、每个文件做备份，而你在用Spotlight搜索的时候，就可以直接利用时间机器来查找以前被删除的文件，并随时恢复。

我们还能通过时间机器查看任何一个文件夹里在每一天的文件变化情况。假如你查看Pro Tools的工程文件夹，那么你就能轻松找到每天每一次录音的波形文件。除了Pro Tools可以帮你做备份，苹果系统现在也可以做备份了，而且恢复起来非常简单轻松！

唯一遗憾的是时间机器需要第二块与现有硬盘完全一样大小的硬盘做备份，但是对备份非常在意的音乐制作来说，这点开销完全不算什么，性能和实用性才是大家最需要的。

虚拟桌面：

没啥好说的，实际第三方的软件已经做的很好了，10.5系统完全模仿着做了一个。

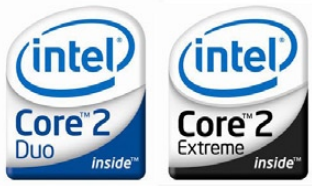
文字到语音：

Text-to-Speech功能以前就有，但说实话很不实用。新改进的版本已经能够提供非常好的发音，相信会成为音乐人的辅助工具。

华丽外表：

虽然从目前Apple提供的新系统截图来看，与以前的系统没什么区别。但苹果保证在明年年初发布时，新系统会有全新的让人惊异的华丽效果！现在不把这些华丽效果展示出来的原因很简单：防止Windows Vista无耻的抄袭。届时Mac OS X 10.5将与Windows Vista几乎在同一时间上市，微软想抄也来不及了呵呵。

PC电脑趋势谈：扣肉全接触



熊猫/文

一、全新的扣肉

7月底，Intel发布了备受期待的基于新一代酷睿微架构的桌面处理器Conroe处理器，性能相比上代产品大幅提升。出于谐音，Conroe处理器被网友们称作“扣肉”，相信大家都或多或少的听到了扣肉的大名，但由于Intel新一代Core 2 Duo处理器变化实在太多太多，很多东西可能大家还不是特别了解，本期我们就一一道来。



应用于桌面的酷睿微架构处理器——“扣肉”

二、扣肉五大创新特性

Intel宽位动态执行（Intel Wide Dynamic Execution）

在Intel的Pentium4和AMD Athlon XP竞争的时代，Intel一直在鼓吹频率之上的法则，但是事实是频率完全不在一个档次的Athlon XP处理器在性能上毫不逊于Pentimu4，Intel前一代的Pentium D已经采用了双核处理器，但是其发热量大的惊人，这也是Intel充分认识到了问题的严重性，因此在“扣肉”处理器的最大特性就在于能效比，即“每瓦特性能”，性能＝频率×每个时钟周期的指令数 是Intel新提出的对性能的创新理解。Intel宽位动态执行的出发点，就是为了提升每个时钟周期完成的指令数升，从而显著改进执行能力和能效。

扣肉就是这样的CPU，在中关村已经面世的扣肉Intel Core 2 Duo E6300 处理器

其主频只有1.86G，而前一代Pentium D的最低主频型号也有2.66G，但扣肉带来的却是更为强劲的性能，高性能低功耗的处理器时代已经到来。

Intel智能内存访问（Intel Smart Memory Access）

Intel智能内存访问技术可以降低内存延迟，提高内存的访问效率。

这项特性的原理就是实现了一种类似于“预处理”的功能，提前对内存读取顺序进行分析，智能、预测性的装载下一条指令所需要的数据，这样能够减少处理器的等待时间减少闲置，同时降低内存读取的延迟，而且它可以侦测出冲突并重新读取正确的资料及重新执行指令，保证运算结果不会出错，大大提高了执行效率。

Intel高级数字媒体增强（Intel Advanced Digital Media Boost）

这项技术就是把前面提到的性能＝频率×每个时钟周期的指令数 这个新概念应用到了多媒体计算方面，进一步增强了计算机的多媒体性能。

Intel智能功率能力（Intel Intelligent Power Capability）

Intel智能功率能力，这种翻译方式虽然生涩，但也很清楚的说明了这项技术的特色，它可以进一步降低功耗，优化电源使用率，从而为计算机提供更高的每瓦特性能。新一代处理器在制造工艺方面做出优化，采用了先进的65nm Strained Silicon技术、加入Low-K Dielectric物质及增加金属层，相比上代90nm制程减少漏电情况达1000倍。

通过该特性，可以智能的打开仅仅是当前需要的子系统，而其他部分则处于休眠状态，这样将大幅降低处理器的功耗及发热。

Intel高级智能高速缓存（Intel Advanced Smart Cache）

AMD曾经指责Intel的Pentium D处理器效率不高，不能算是真正意义的双核处理器，这是因为Pentium D的缓存系统是独立于内核的，因此两个核心都只能通过相对低速的前端总线才能访问到另一个缓存系统，因此造成了低效问题。

而Intel酷睿微结构体系结构，采用了共享二级缓存的做法。CPU读取的时候不经过前端总线，直接可以访问到全部的缓存地址，大大增强了效率。

相当于一个工厂有两个仓库，扣肉的两个仓库是连通的，而Pentium D的两个仓库隔着一条街道，因此扣肉要提货就直接提了，而Pentium D要用车从街道的另一边把货载过来。

三、扣肉带来的影响

奔腾成为历史 桌面、移动和服务器架构统一

近十年的时间，人们都把奔腾(Pentium)视为Intel CPU的代名词，而如今酷睿微架构的到来告诉我们奔腾时代已经过去。

在此之前，Intel针对桌面、移动和服务器市场的处理器，是基于不同的架构，而今，Intel第一次在所有平台上使用了统一的构架，即Core微体系架构，其针对桌面、笔记本和服务器推出的产品代号分别是，Conroe（就是我们说的“扣肉”）、Merom和Woodcrest，都拥有64位处理能力，并且都是双核产品。

性能≠频率

尽管AMD早就已经向人们证明了CPU不依靠高频率就能完成人们的需求，就能和Intel相抗衡，但是直到业界老大Intel也发布了低频的扣肉之后，人们的观点才完全转变了过来，CPU主频已经不再是衡量性能的第一指标了。

四、扣肉 VS AM2

就在扣肉上市的同时，AMD的AM2系列处理器疯狂降价，其中AM2 Athlon64 3800+X2已经从2XXX元降到了1200元，实在是有着非常高的性价比。

从技术和制造工艺上比较，扣肉是绝对处于领先地位的，Intel摒弃了沿用多年的NetBurst架构，让扣肉脱胎换骨。而AM2实际上只是增加了对DDR2内存的支持；从实际应用来看相同价位的扣肉E6300和AM2 4200+X2 的效率基本持平，E6300稍有领先，不过需要注意的是扣肉才刚刚上市，有很大的降价余地，真正与其竞争的应该是3800+X2，因此，可以看出扣肉在绝对实力上胜出了一筹。

从性价比上考虑，购买AM2双核处理器是超值的，但是支持AM2的主板芯片组NForce 570 系列有一定的内存兼容问题。因此如果您缺少装机经验，请谨慎考虑。

五、把扣肉端回家

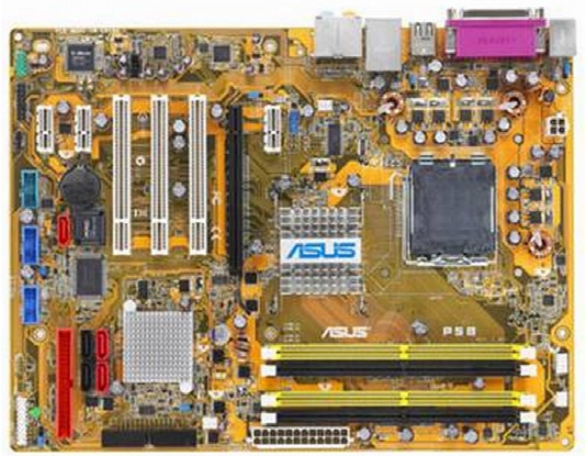
现在客观的讲并不是购买扣肉的最佳时机，随着各个型号的不断发布，和来自AM2降价的压力，扣肉势必会降低价格，并取代如今Pentium D的市场。而AMD的几款双核CPU现在都有着极高的性价比。

为了急于尝鲜的朋友，我们还是看一看当前扣肉的市场状况。

我们先来看一下扣肉的价格，目前已经能在中关村买到的Intel Core 2 Duo E6300 1.86GHz的报价为1600元左右，当然这是一款高端CPU的价格。比较现在热销的AMD AM2 3800+还是缺少了一些性价比。

想把扣肉端回家还一定要为其选个合适的“盘子”，也就是选择一块性能强劲稳定的主板，不然把扣肉打翻了就损失惨重了。

目前最适合与扣肉搭配的桌面级主板芯片组是Intel P965，主板一线大厂都已经推出了P965芯片组的主板：华硕P5B 售价1499元、技嘉965P-S3 售价1188元、微星P965 NEO 售价1100元。



华硕P5B

大家需要注意的是，目前有一些小品牌的主板厂商已经推出了价格较低的P965主板，尽管扣肉为低功耗的CPU，但它对主板的供电要求更为严格，因此大家选购的时候，尽量选择一线大厂的主板与之搭配。给扣肉找一个放心的盘子。

征稿启事

■ 新闻视点：新闻事件深入分析。

■ 独门秘籍：电脑音乐制作、编曲、录音、缩混、声音合成、音响等方面的技术文章，在第一时间学习高手的高招。

■ 抢先评测：最新软件/硬件产品评测，拒绝老产品老面孔登场。

■ 对决PK台：挑选几个同类软件或硬件产品做横向评比，看清每个产品的优缺点，更要看清哪个产品更适合你。

■ 视频教程：以最直观的视频形式，手把手的教您制作音乐，必须在线观看。

■ 小贴士：Cubase SX/Nuendo、Sonar、Pro Tools、FL Studio、Reason、PC电脑、苹果电脑的各种小技巧和新消息。这7个小栏目是每期必不可少的保留节目。

■ 新棚友：每月新加入的录音棚/工作室照片和信息。

稿件要求：

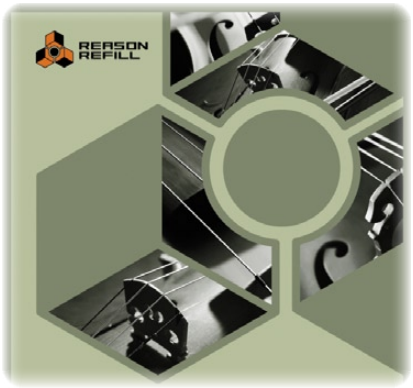
来稿必须保证是作者原创，如有抄袭剽窃等侵权问题，本刊概不负责。向本刊投稿者不得一稿多投。本刊对来稿有删节权,不同意删改者请注明。

虽然杂志是免费发行的，但作者均可得到稿费。来稿一经采用，稿费从优，普通稿件稿费 80元/千字，长期供稿可作为签约作者，稿费 100元/千字，稿费于杂志发行1周后结清。

为确保我们能及时收到您的稿件，请将来稿发送至musiXboy的信箱：mx@midifan.com，或直接通过论坛短消息联系musiXboy。

点这里进入论坛讨论此文

正确调节鼓loop的音高



Derek Johnson/文　musiXboy/编译
(Sound On Sound杂志 2005年12月期)

一直不能让Reason的鼓节奏与你的旋律很好的结合在一起吗？如果你从来没考虑调节鼓采样或鼓loop的音高，来创造与旋律更融合的混音，那么你就要仔细看看这篇文章了。

在编曲的时候，许多朋友都是简单的把鼓loop或鼓采样与其它旋律简单的堆叠在一起，你可曾想过，哪个声部使得整个混音变得不够协调？有多少人去想过鼓声音的音高平衡问题？

所有的采样音色都有其固有的音高，就算那些我们认为是杂音的非音高乐器（比如下雨声采样）也是一样的，但是我们不必去关心那些非音高乐器的实际音高，我们关心的是非音高乐器与其它带音高乐器之间的音高协调。这对于你最终的混音非常重要，要知道在录音棚里每个鼓手在录音前都会根据音乐的实际情况调节鼓的音高。

如何在Reason里调节鼓采样的音高？

在ReDrum鼓采样器里，每个鼓采样通道都有一个Pitch音高参数，可调范围在原始音高的正负八度以内，虽然没有音高微调，但也足够用的了。在10个通道中，第6、7通道比较特殊，除了可调音高正负八度，还可调bend弯音参数，这样我们就能做出类似合成器的那种带弯音的鼓声音。



ReDrum鼓采样器第6、7、8通道的音高调节

其它可以读鼓采样的模块就是NN19和NN-XT采样器了。在采样器的震荡器部分，OCT旋钮可以调节正负八度，SEMI旋钮调节半音，FINE旋钮则以音分为单位进行音高微调。



NN19采样器的音高调节

正确调节鼓采样音高：

1．我们拿底鼓来举例吧。首先将一个底鼓采样读取到ReDrum鼓机里（或者用NN19、NN-XT采样器也成），然后播放你的bassline。

2．将主旋律设置为循环播放区域。比如你的bassline是降E调的，那就选1-2小节包含大部分降E音符的区域进行循环播放。

3．然后试着调节底鼓的音高，你不必让底鼓的音高完全变成降E调，只要底鼓听起来和bassline很和谐就可以了，也就是听起来二者不要发生冲突。

4．如果你听不出怎么叫和谐怎么叫不

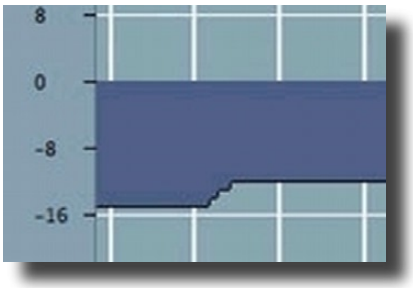
冲突，那就用弦乐调音准的小技巧来做：把音高旋钮拧到最大或者最小，然后慢慢向中间靠拢，直到你感觉和谐为止。

5．在1-2小节内找到合适的底鼓音高之后，让底鼓和整个混音一起播放。虽然大部分的底鼓不会与混音中的高频发生冲突，但还是检查一下为好。

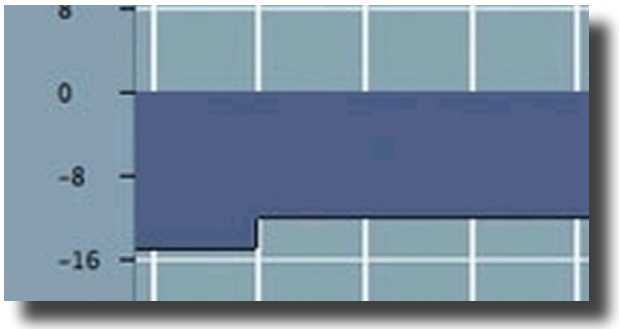


为不同主音的旋律使用不同音高的鼓
通过pitch参数的automation实现

如果整个音乐在中间有转调，那么在转调之后最好重新调节一下底鼓的音高。大多数情况底鼓不必做很猛的转调。记住，你没必要让底鼓去适应所有的主音，只要听起来够和谐就好了。如果底鼓与一个声部听起来很和谐，但跟另外一个声部听起来就非常不和谐，而你又找不到一个使所有声部都和谐的底鼓音高，唯一的解决办法就是：确定一个让二者单独听起来都不和谐的底鼓音高。



你可以手动录制鼓音高的automation
找到合适的音高



之后再对automation曲线进行修正

正确调节鼓loop音高：

除了鼓采样，在Reason里我们还经常使用Dr.REX模块去读取现成的鼓loop。主要思想与前面讲的调节鼓采样一样，我就不重复了。



使用Dr.REX可以方便的调节
整个loop或每个切片的音高

Dr.REX模块通过OCT旋钮可以以八度为单位调节整个鼓loop的音高，FINE旋钮则是微调，ENV AMOUT跟我们今天讲的关系不大，它使音高包络的跟随度发生变化，产生弯音效果。

你也可以很简单的调节每个切片的音高，选中某个切片，转动PITCH旋钮就可以了。但是你无法做到微调，解决办法：按照切片时间，制作快速变化的FINE旋钮automation，不过这确实是个精细活。



www.midifan.com