

也许，你正在经历着这些困扰

总有没完没了的bug要修

总在DDL最后一刻，又收到一大堆火急火燎的新需求、并被要求一定要通宵搞完

多人协作的工作表、WWU、Originals数据总是乱七八糟

创建、重命名、拖拽、勾选、点击、按住Ctrl、按住Shift、切换Tab、切回刚才的Tab、新建文件夹、文件夹里再新建文件夹……无限轮回

音频需求的优先级总被排到最低，总要给其他需求让路，却又总被拖后腿

挂接总要配一大堆组件、翻一大堆蓝图、开一大堆编辑器、画一大堆圈圈、建一大堆轨道、打一大堆关键帧……

好不容易挂接好了，提交数据时一对比，发现产生了一大片不明差异……

测试版本好不容易构建好了，终于可以跑测了，开十次，崩九次

bug单总是只有一个光秃秃的视频和一行字，总要再重新花时间开版本、崩版本、卡版本、输无数个GM指令、刷无数条Log……

渐渐地，桌面上堆满了Excel表、PPT、Word文档、素材库、外包资源、需求表、挂接工具、跑测工具……

但好像，某个文件夹深处还屯着几个买了一年用过两回的插件，角落里还躺着一个正在吃灰的Midi键盘……

掐表算了算，每天有几个小时在“做创作”，又有几个小时在“做杂务”？

时间都去哪儿了？

也许你早已习惯了这些“隐形杂务”，甚至觉得“理所应当”

如果总是疲于奔命，那人生可要错过太多有趣的风景了，不是么？

Wwise闪电分身 v2.0 主要功能简介



一键初始化工程

对于尚未产生过WWISE数据的新工程，插件中包含了一些预设好的内容，可直接一键生成初始的WWISE工程！设计师不必再花时间创建那些几乎所有项目类型都有的对象。

一键整合、镜像、另起炉灶

命名即所有。在设计师决定好命名的那一刻，工程内，从容器框架、指派通路、导入资源到创建Event、归属Bank等一系列的过程，一键自动完成！

设计师无需在“拖拽点击”上浪费过多时间（尤其不用为了套用某个模板结构，再花时间在本地新建许多层文件夹），从而可以专注于设计本身的打磨和动态混音，把精力用在刀刃上。

一键扩展Switch对象

当某个Switch组增加了新的Switch对象，随之，许多相关的容器及资源也都需要增补时，可以通过一键扩展Switch内容来完成扩充。设计师无需为大量容器逐个手动添加指派。

一键全局安全检查

是否存在未经指派到Event的容器对象？是否存在忘记导入到容器的闲置wav文件？是否存在未被使用的闲置Event事件？是否存在未遵守结构规范的对象？……

在工程结构严格遵守规范的前提下，设计师可一键获得以上所有易错项的检查报告，无需担忧人为疏漏。

一键重联wav

由于wav的本地位置发生变更导致的Sound原索引失联，可一键自动重连！

一键导出需求表

当需要第三方供应商量产样本时，可以在选中需要处理的对象后，右键一键导出需求表。

一键转工作表

在符合格式规范的前提下，老工程的Event数据可一键导出插件可读的JSON文本。

插件的JSON，也可一键转xlsx，用于团队协作；若老工程的数据是在xlsx中管理的，也可将xlsx格式化后，一键转为插件可用的JSON，由插件来接棒管理！

一键播放、停止、展开Event

在数据表相应单元格上，右键“播放、停止、展开Event”，可一键快速验证声音反馈和查看细节。设计师无需再回到Wwise界面，花时间搜索出Event、点击走带播放等等。



免责声明

本插件（Wwise闪电分身 v2.0）仅供测试研究和专业交流探讨之用。

开发者暂时无法为您担保使用过程中完全不会出现任何形式的数据丢失、数据损坏的情况。

如果您使用本插件，则意味着您愿意自行承担使用过程中任何维度的风险以及可能造成的损失、且不需要开发者承担任何责任。

如果您符合以下任意一种情况，本插件大概率能够帮助到您

我的工作是“游戏音频设计师”、“游戏音频PM”、“游戏音频策划”、或“游戏音频程序”等游戏音频专业相关。

我使用过Wwise，了解Wwise的基本操作流程，工作中也正在使用Wwise。

我的创作环境苛刻、日常任务压力、时间压力大，身边也没有程序员辅助，想试试看这个插件能不能帮自己节省一些时间。

我想专心做内容设计和创作，不想在各种各样的工具和操作流程上耗费太多时间。

我希望通过观察和了解插件本身的形态，获得一些关于“流程优化”“效率提升”“数据管理”“工具开发”等方面的思考或启发。

我正在学习游戏音频相关的知识，且正在学习和练习使用Wwise，希望通过这个插件获得一些学习上的辅助。

我正在跟进一个新的游戏项目，有充分的时间和机会部署音频基建工作，想把千丝万缕的流程和数据，在短时间内打理得清清楚楚明明白白。

如果您符合以下任意一种情况，这个插件可能并不适合您

我的工作并不是“游戏音频设计师”、“游戏音频PM”、“游戏音频策划”、或“游戏音频程序”中的任意一种，也对游戏音频行业没什么兴趣。

我的工作中并不使用类似Wwise的音频中间件，也不打算学习和了解相关专业知识。

我大部分的工作量都依靠着“繁杂的人工操作过程”来支撑，担心自动化和AI会让自己的价值削弱或失去价值。

我在日常工作中并不介意、且允许有“繁琐的操作过程”，甚或其实比较享受“繁琐的操作过程”。

我手里的项目数据已经混乱到无力回天，并且我已经身心俱疲，不想再投入任何精力做所谓的维护、管理、优化。

我担心本插件可能会造成任何数据上的损失，比如数据损坏、数据丢失、数据混乱等。

我担心本插件可能会带来任何精神上的不愉悦，比如压力感、焦虑感、失落感、恐惧感等消极感受。



版权声明

本插件（Wwise闪电分身 v2.0）相关所有文件数据，包括且不限于代码、可执行程序等，构成成分复杂，或涉及多种版权许可证类型。
目前除了“**纯个人研究**”、“**非商业的、注明来源的分享交流**”之外，开发者暂时无法通过本声明直接向您个人、或公开提供任何用于其他用途的授权。

如果您期望将本插件（包括任何相关数据）用于其他用途，请务必先通过[开发者邮箱](#)联系我，在确认好您的用途不会给您自己带来麻烦后，再进行使用。
开发者将不为您的任何使用行为承担任何责任。

注：为了给促进行业发展增添一份绵薄之力、支持分享互助的开源精神，开发者正在努力推进版权许可相关事宜，力争后续能够将插件的源代码尽可能以自由度最大的许可证授权类型公开发布，让感兴趣的小伙伴们都能参与到二次开发和接力中！时间不会太久，请您耐心等待！

隐私声明

本插件（Wwise闪电分身 v2.0）的运行不依赖网络。当您使用时，本插件不会自动收集任何您在计算机上的隐私信息。

开发者邮箱

whiphelper@gmail.com

Wwise闪电分身 v2.0 说明书目录

[主要功能简介](#)

[免责声明；版权声明、隐私声明和开发者邮箱](#)

[不同WWISE版本的适配情况](#)

[如何安装](#)

操作示例

[启动插件](#)

[初始化工程](#)

[一键整合！如何整合一个单纯的UI点击声](#)

[一键整合！如何整合一套玩家的脚步声](#)

[一键整合！我有一个自定义的模板结构，应用广泛，但结构比较特殊，插件的三个预置基础模板框架无法应对，怎么办？](#)

[关于命名规范](#)

[一劳永逸！配置自动化模板！理解插件的核心概念](#)

[一键扩展！地面前置增加了！有许多涉及材质Switch的容器都需要扩充内容，怎么办？](#)

[一键镜像！想完全复用一个Event，但从里到外的容器、资源命名，都得是全新的！怎么办？](#)

[一键重置！另起炉灶！AI上一轮生成的效果一般，再来一遍！](#)

[一键扫描！全量安全检查！](#)

[一键重联！WAV索引失联！怎么办？](#)

[一键整合！需要Loop的声音对象，一键组团！永不遗漏！](#)

[一键导出！如何一键导出xlsx需求表？](#)

其他辅助功能和使用技巧

[音频简表导出；自动打Bank；全量安全检查后Event上色；常规偏好设置；快速验证Event状态；Event一键更新Notes；检查替条；](#)

[另存JSON和xlsx；xlsx转JSON；基于工程内容转JSON；检查音频ID挂接状态；过滤器；搜索文本；设置颜色；版本管理；报错说明](#)

开发者寄语



不同WWISE版本的适配情况

*当前版本仅支持Windows 10 和 Windows 11

	一键初始化工程	一键整合
2022.1.0 及以上	完全支持	
2021.1.1 ~ 2021.1.10	Assign RTPC到BUS 不支持 Create Meter Effect不支持	完全支持
2019.2.8 ~ 2021.1.0	Set RTPC Value 不支持 Assign RTPC到BUS 不支持 Create Meter Effect不支持	完全支持
2019.1.0 ~ 2019.2.7	Set RTPC Value 不支持 Assign RTPC到BUS 不支持 Create Meter Effect不支持	Auto Color 不支持
2018.1.2 ~ 2018.1.11	Set RTPC Value 不支持 Assign RTPC到BUS 不支持 Create Meter Effect不支持	Auto Color 不支持 Set Pitch Random 不支持
2017.2.10 ~ 2018.1.1	Set RTPC Value 不支持 Assign RTPC到BUS 不支持 Create Meter Effect不支持 自动创建WorkUnit不支持，必须先手动创建WorkUnit， 将路径信息输入插件后，后续部分才能自动化	Auto Color 不支持 Set Pitch Random 不支持
2017.2.10 以下版本	完全不支持	完全不支持

- Set RTPC Value 不支持：指插件通过Waapi创建新的RTPC对象时，无法直接初始化其最大值、最小值、默认值
- Auto Color不支持：指通过插件的安全检查功能筛查出有问题的对象时，无法通过Waapi直接修改其在工程中的图标颜色，以给到设计师直观的视觉提示
- Set Pitch Random不支持：指插件无法直接通过Waapi给Sound对象的Pitch自动开启并应用预设的Random值
- 自动创建WorkUnit不支持：指插件无法通过Waapi在工程中自动生成WorkUnit并创建相应的WWU
- Assign RTPC到BUS 不支持：指插件无法通过Waapi在工程中给BUS上指派RTPC对象
- Create Meter Effect不支持：指插件无法通过Waapi在Effects路径下创建Meter

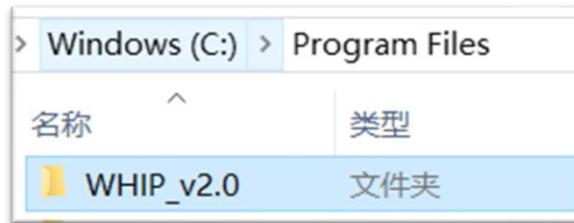


如何安装

* 方法A：对于 WWISE 2021.1.10以下的版本，只能用下面的方式安装

第一步：

将插件的文件夹“WHIP”拷贝放置在硬盘的任意位置（比如，C:\Program Files目录下）。



第二步：

打开插件文件夹，鼠标右键点击Launch.exe，在右键菜单中点击“发送到”，再点选“桌面快捷方式”。以后使用时，先打开Wwise工程，再双击这个快捷方式即可启动插件。



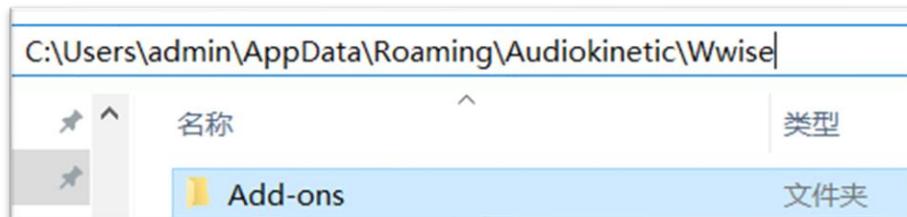
* 方法B：对于 WWISE 2021.1.10及以上版本，**建议使用下面的方式安装**

第一步：

将Add-ons文件夹放置在该路径下，即完成安装：

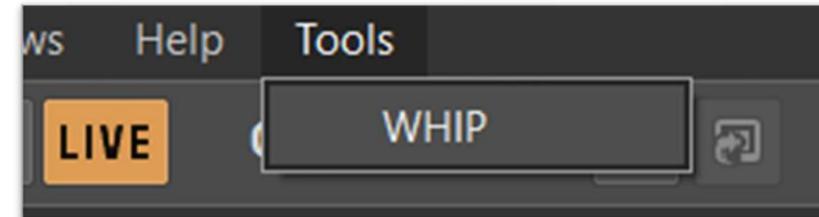
C:\Users\用户账户文件夹\AppData\Roaming\audiokinetic\Wwise

注：如果在移动的过程中遇到冲突提示，请立即取消移动（不要强行覆盖任何数据），并改为采用方法A安装！如果过程中仍然出现冲突，请立即停止一切操作，并放弃使用本插件。



第二步：

开启Wwise后，菜单栏顶部出现一个Tools。点击“Tools”菜单，在下拉菜单中点选“WHIP”即可启动插件。



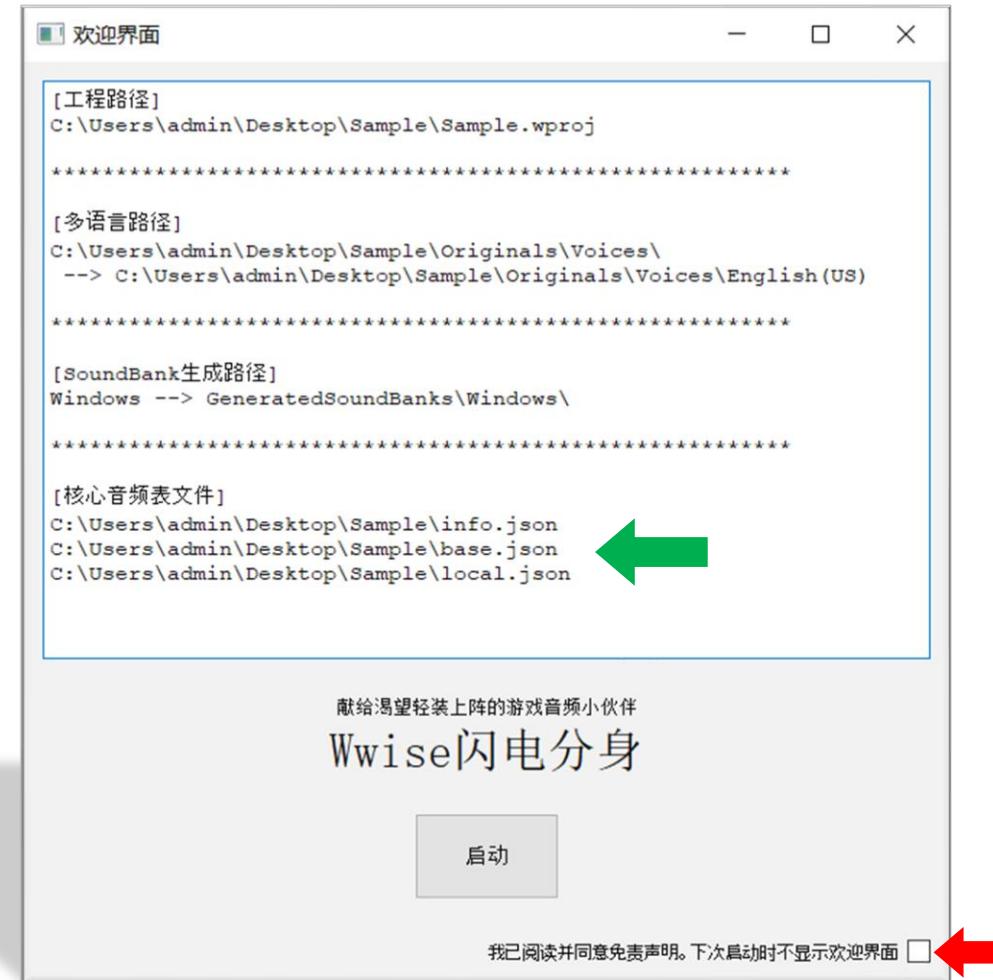
启动插件（首次）



首先，为确保数据安全，插件只能在有且仅有一个Wwise工程处于开启的状态下运行。

如果你看到欢迎界面，且启动按钮处于可点击的状态，说明插件已检测到工程、以及在工程路径下生成了3个必要的核心JSON文件，并可以正常启动。

如果首次启动时勾选过右下角不显示欢迎界面的勾选框，则后续的启动，都将直接进入主界面。



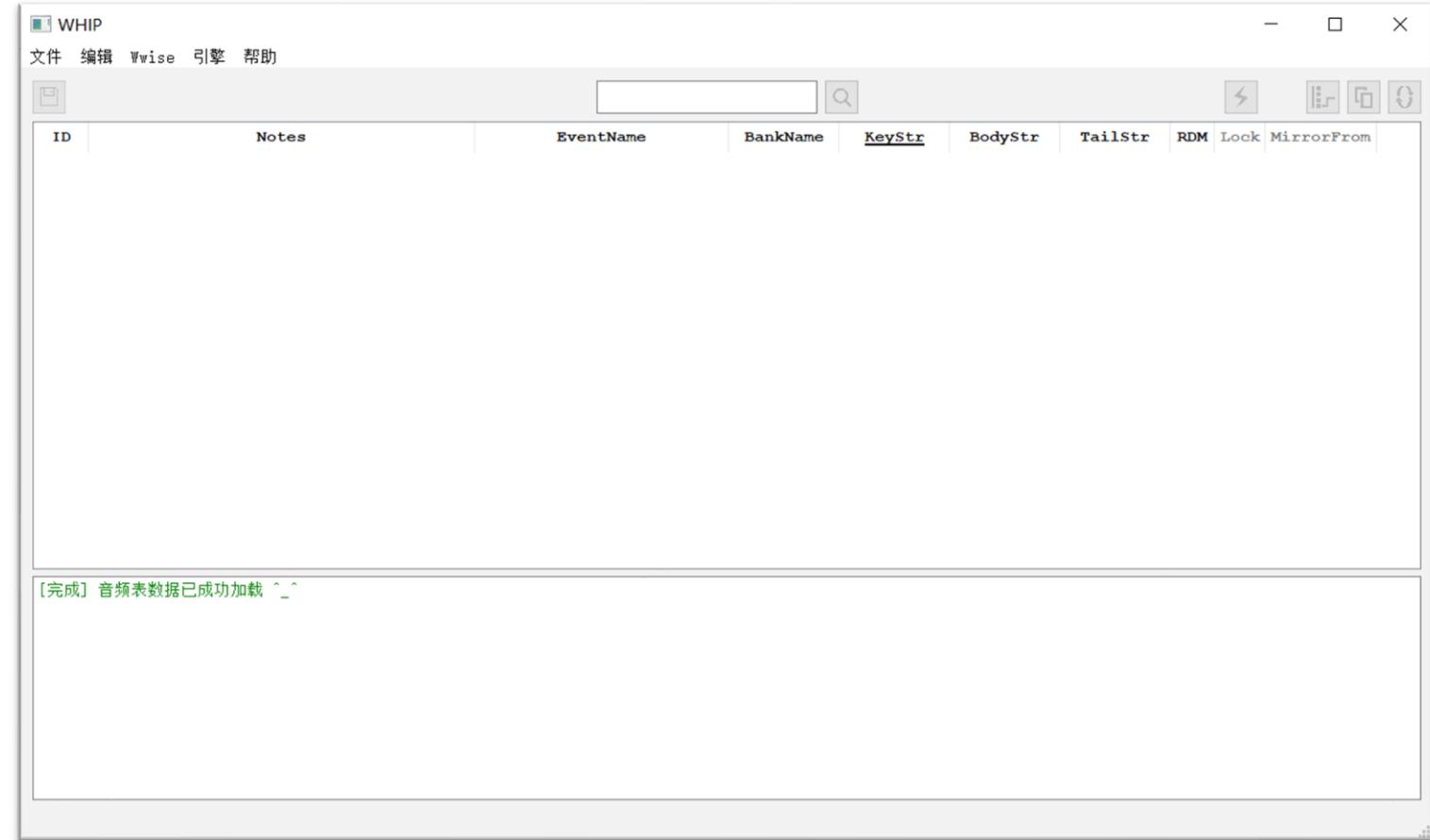
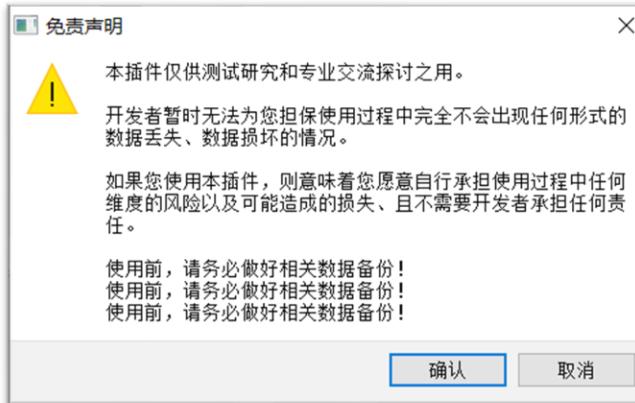
温馨提示：

“info.json” “base.json”
“local.json” 三个文件中，记录着插件中大部分的数据信息，是插件的“核心”和“本体”。

启动插件（首次）



点击启动按钮后，可以看到“免责声明”提示。如果点击确定，则会直接进入插件的主界面。

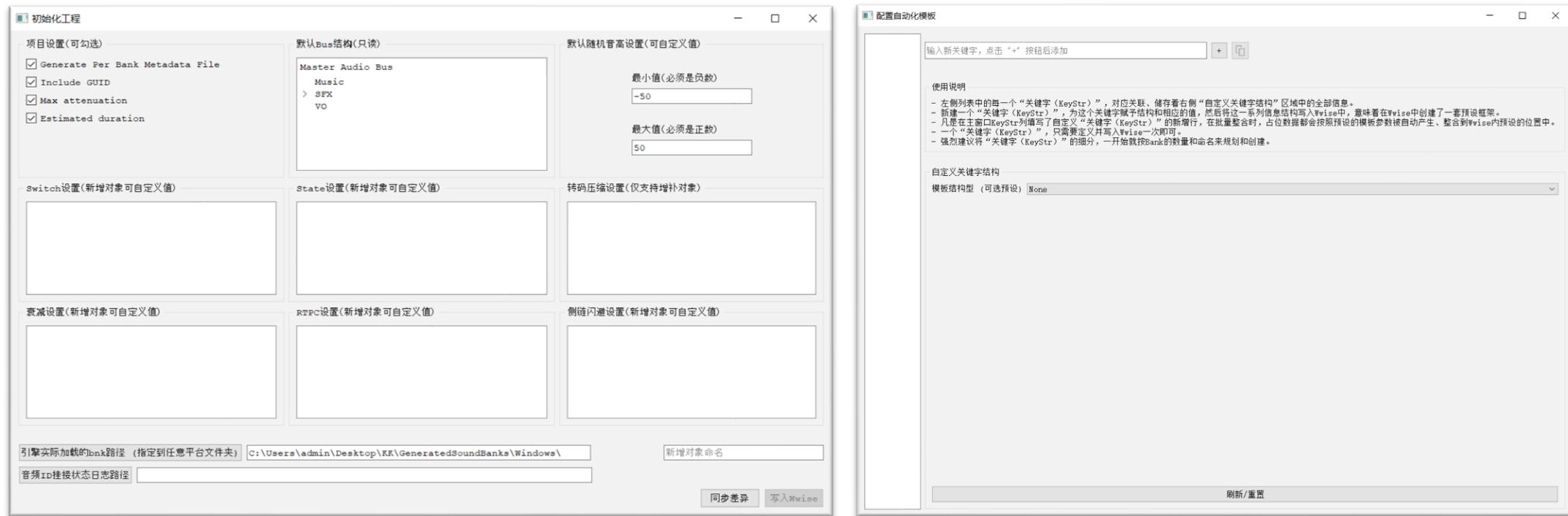


启动插件（首次）



插件除了主界面之外，还有另外两个界面：初始化工程、配置自动化模板。

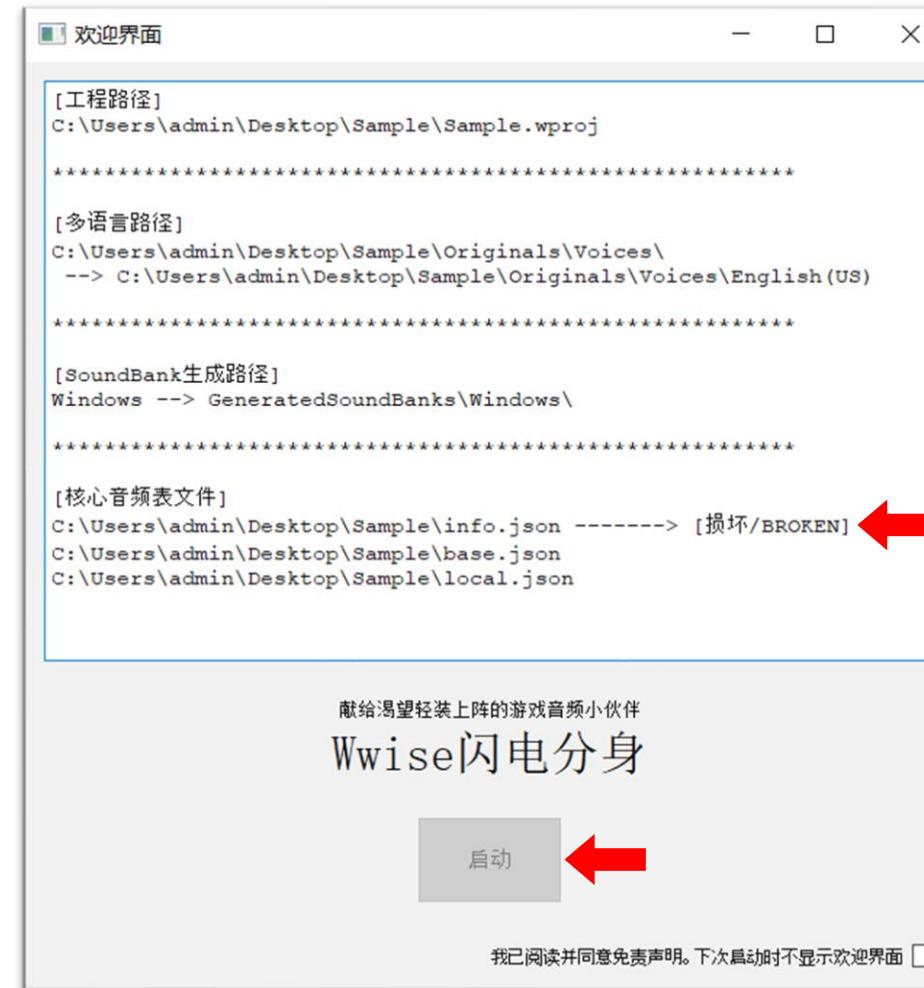
不过，这两个界面并不常用。主界面中的“表格”才是插件的主要操作区域。



启动插件（非首次）



如果启动按钮处于灰色的禁用状态，说明插件未找到核心音频表文件，或文件有损坏。
解决方法：修复或删除Wwise工程路径下提示损坏的JSON文件，并重新尝试启动。



初始化工程（首次）



插件中预备了一些常用设置，如下图所示。

如果当前Wwise工程是一个[全新的工程](#)，我们则可以选择需要的内容，一键写入工程中，省去逐个手动创建的麻烦。

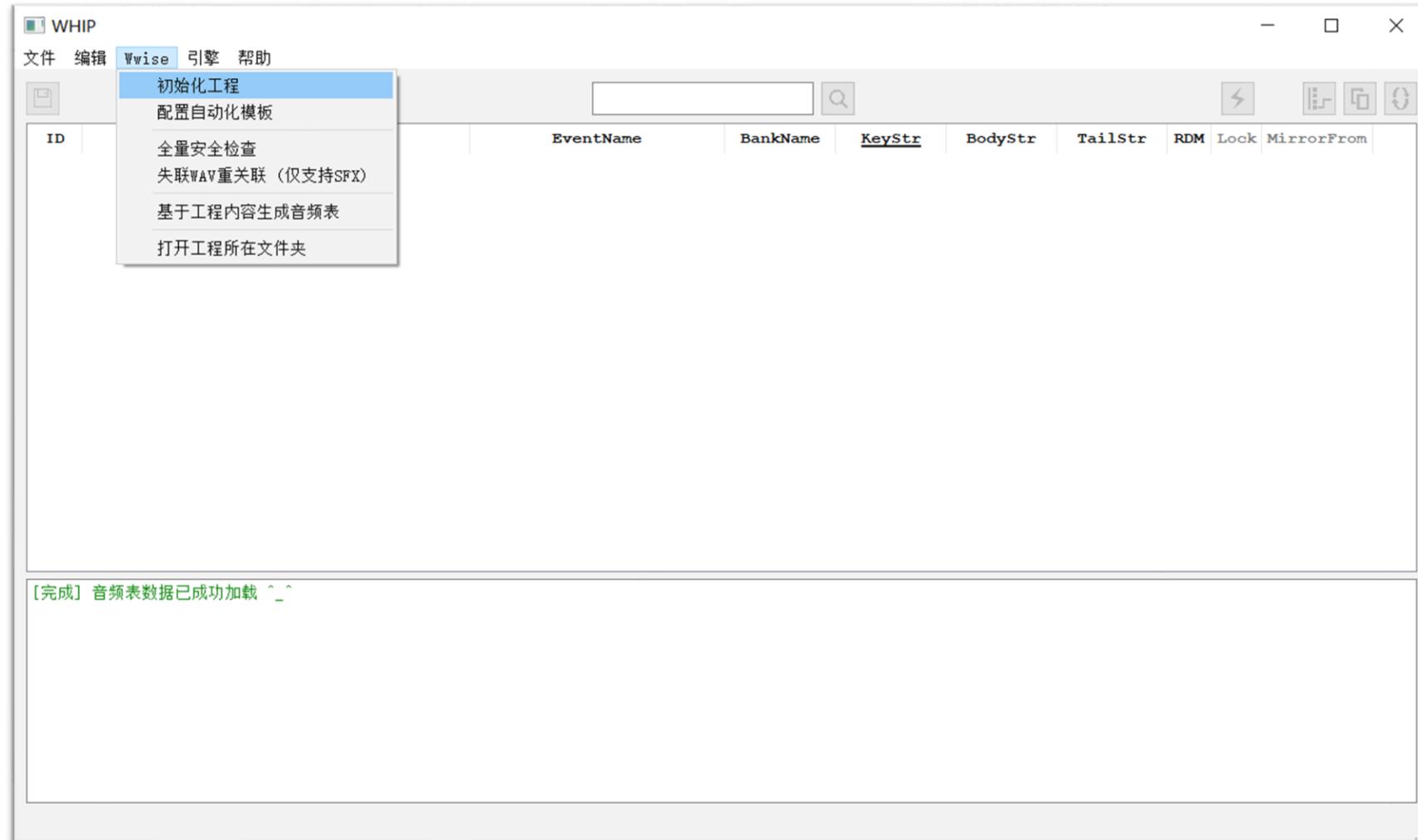
The screenshot displays five panels from the Wwise Work Unit Manager interface, illustrating pre-configured settings:

- Master-Mixer Hierarchy:** Shows a tree structure under "Default Work Unit" including "Master Audio Bus" (with branches for "CG", "Music", "SFX", "Action" (with "NPC" and "PC"), "Amb", "UI", and "VO").
- Switches:** Shows a list of switches categorized by texture type: "Dirt", "Grass", "Gravel", "Metal", "Muddy", "Snow", "Stone", "Water_Aンkle", "Water_Knee", "Water_Puddle", "Wood", "Switch_Impact_Texture", "Concrete", "Debris", "Flesh_Beast", "Flesh_Human", "Metal_Hollow", "Metal_Solid", "Plant_Soft", "Plant_Solid", "Water", "Wood_Hollow", "Wood_Solid", "Switch_PC_NPC", "NPC", and "PC".
- States:** Shows state configurations under "Default Work Unit": "State_Alive_Dead" (with "Alive" and "Dead" states), "State_Playing_Pause" (with "Pause" and "Playing" states), and "Game Parameters" (with "RTPC_HP", "RTPC_SC_Hit_Cast", "RTPC_SC_PCBus_NPCBus", "RTPC_Time", "RTPC_Volume_Master", "RTPC_Volume_Music", "RTPC_Volume_SFX", and "RTPC_Volume_VO").
- Attenuations:** Shows attenuation settings under "Default Work Unit": "Att_Ally", "Att_Enemy", and "Att_Gen".
- Conversion Settings:** Shows conversion settings under "Default Work Unit": "Conv(CG)", "Conv(Music)", "Conv(SFX_NPC)", "Conv(SFX_PC)", "Conv(VO_NPC)", and "Conv(VO_PC)".
- Actor-Mixer Hierarchy:** Shows a tree structure under "Default Work Unit" including "UserDefinedTemplate".

初始化工程（首次）



点击工具栏中的“Wwise”，在下拉菜单中点击“初始化工程”。



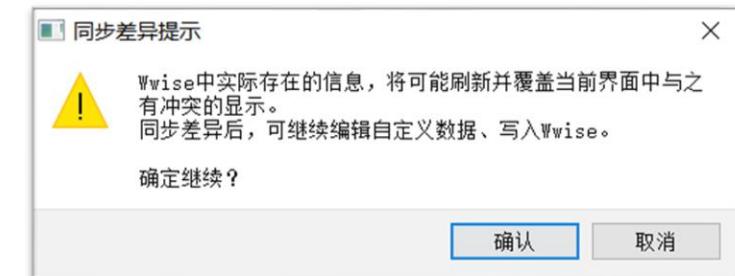
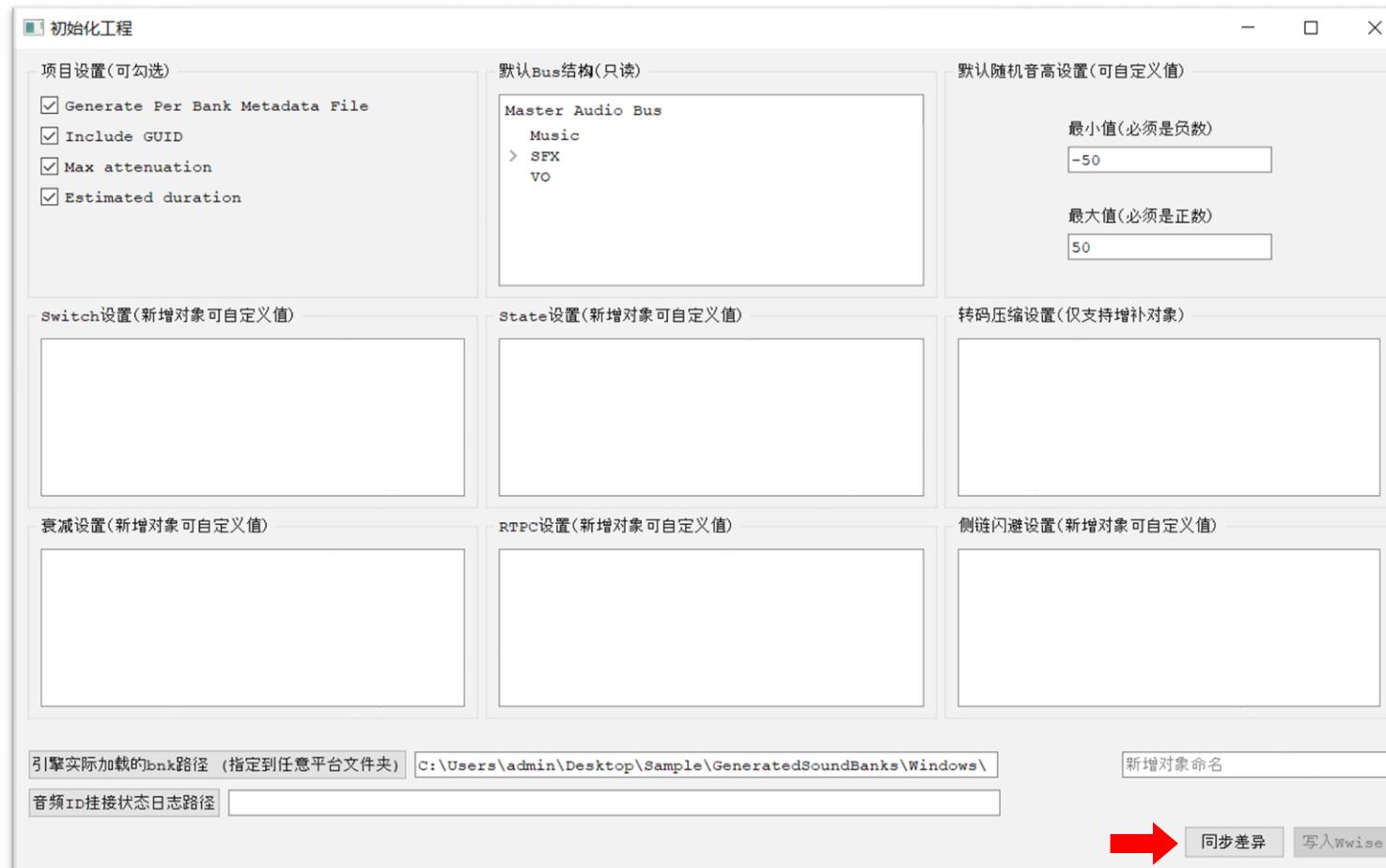
初始化工程（首次）



初始化工程窗口中，有许多个模块区域。

在添加内容之前，需要先点击右下角的“同步差异”按钮，将当前Wwise中的实际情况，同步到窗口中。

点击“同步差异”按钮后，阅读“同步差异提示”弹窗中的信息。确认后，将可以开始编辑预备写入Wwise的数据。

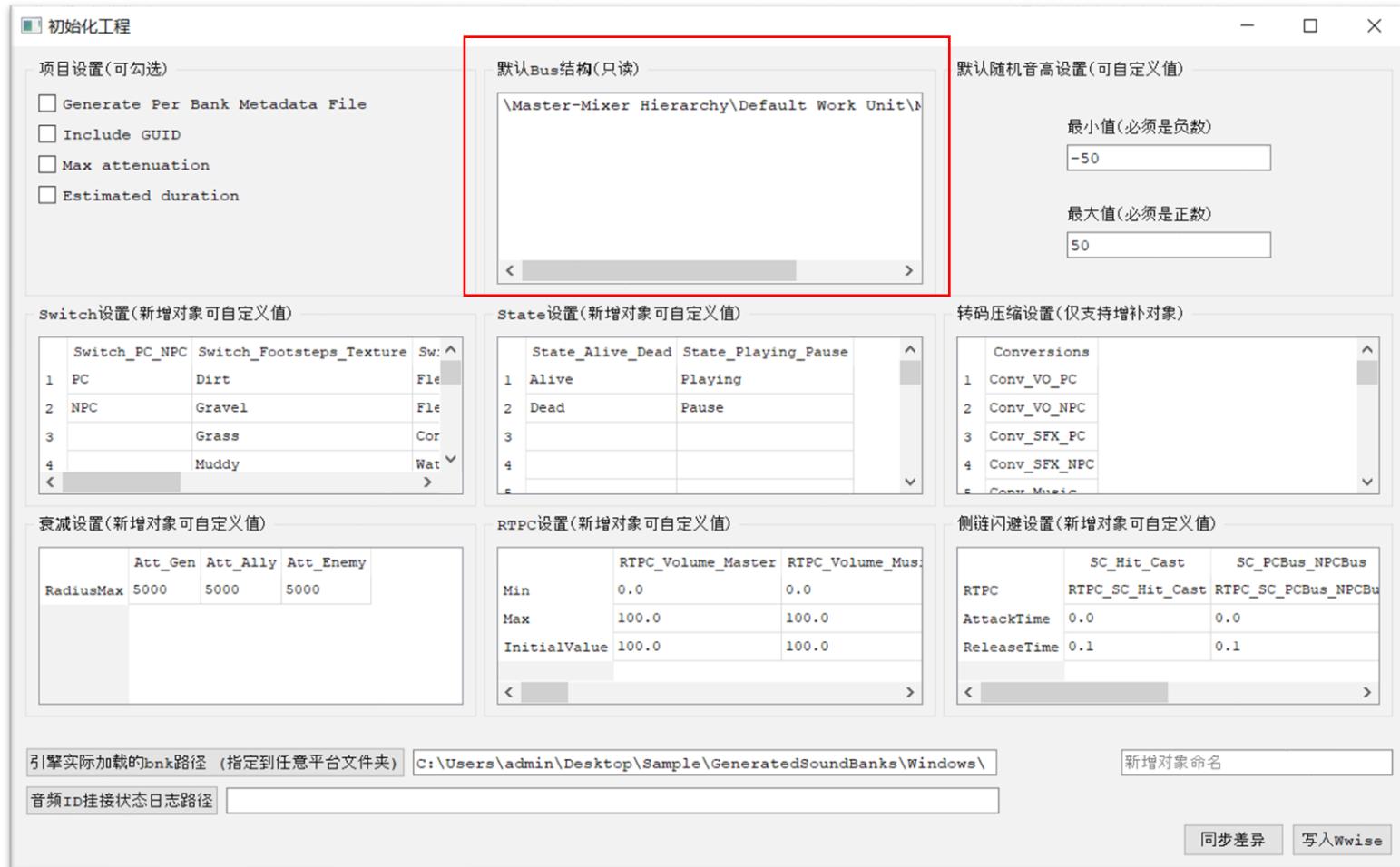


初始化工程（首次）



此时，我们可以看到插件中预先准备的内容显示了出来。

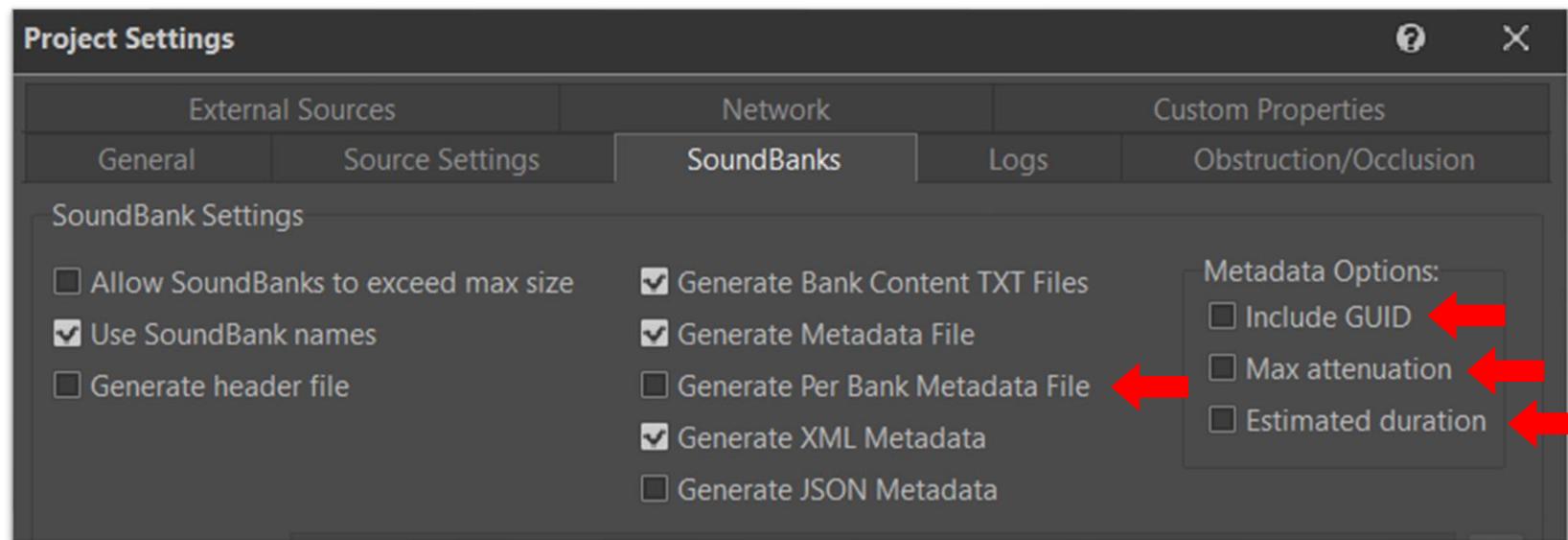
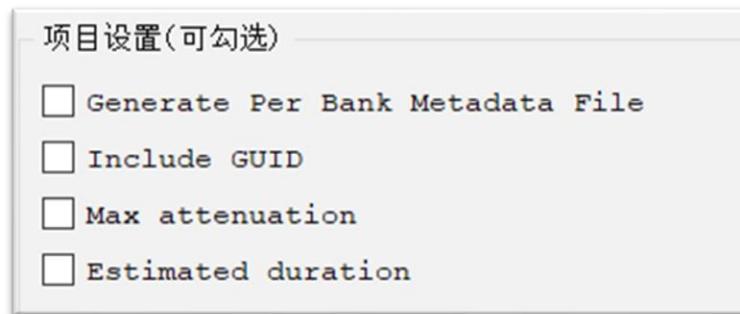
除了“默认Bus结构”区域中的内容不可编辑之外，其他区域的内容都可以根据自己的需要，进行新增、删除、修改。



初始化工程（首次）



“项目设置”区域，对应Wwise工程Project Settings中，SoundBank选项卡里的部分内容。



初始化工程（首次）



“默认随机音高设置”区域中填写的两个值，将作为工具中自动化整合功能为Sound对象默认应用的值。

[\(示例参考\)](#)

默认随机音高设置(可自定义值)

最小值(必须是负数)

最大值(必须是正数)

Name	Value
Enabled	<input checked="" type="checkbox"/>
Min Offset	<input type="text" value="-50"/>
Max Offset	<input type="text" value="50"/>

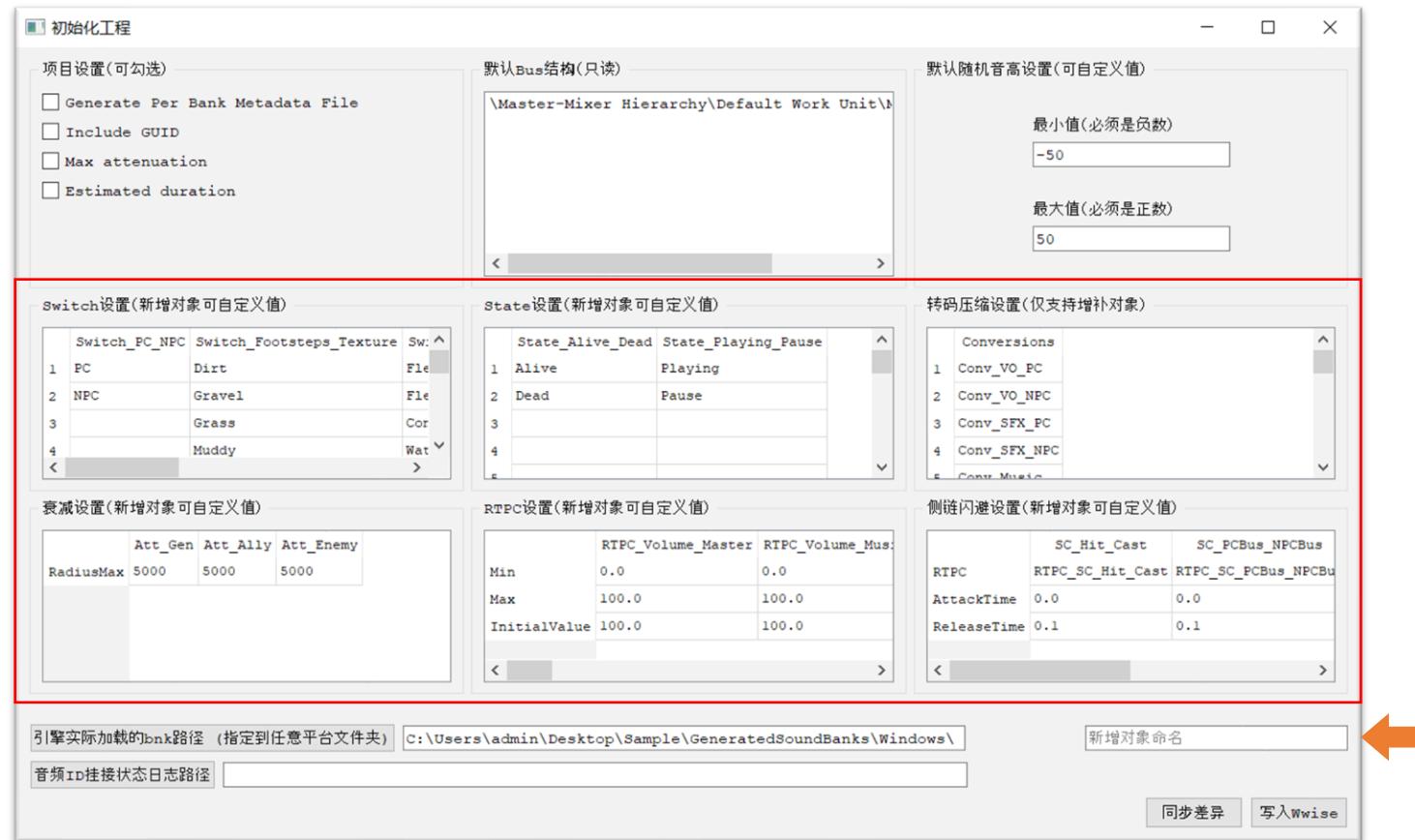
初始化工程（首次）



其余6个区域，依次对应的是工程里Switches、States、Conversion Settings、Attenuations、Game Parameters、Effects(Meter) 的路径。

如果需要在某区域中添加列，需要先在右下角的文本框中输入需要新增的对象名，然后在目标表格中点击鼠标右键，点选“新增列”。
如果需要在某区域中删除列，直接在该区域中需要删除的列的任意一个单元格上点击鼠标右键，点选“删除列”即可。

注：“转码压缩设置”区域中，只能添加或删除单元格内的文本，不能编辑行列。



初始化工程（首次）



最下方还有两个路径信息。

引擎实际加载的bnk路径：

由于在实际的项目中，Wwise打出来的bnk，通常需要放在某个引擎目录下，与游戏其他数据最终一同打包。
这里需要手动指派的bnk路径，就是引擎实际读取的路径，而不是Wwise工程路径下默认的GeneratedSoundBanks路径。

注：在主界面中有一个右键功能是“填充Bank信息”，目的是为了确认和定位某个Event实际存在于哪个bnk中。该功能的查询路径就是这里的路径。

音频ID挂接状态日志路径：

有一个[未包含在本插件中的功能模块](#)，可以将配置在引擎中、技能编辑器、动作编辑器、地图编辑器、脚本、表格等各个地方的音频ID，与本插件音频总表中的所有音频ID做一个全量的对比，将两边不匹配的音频ID及其所在位置，整理到一个JSON文件中，用于QA，也给设计师起到提示的作用。
这里的路径，指的是通过引擎端一个额外开发的功能导出的对比结果JSON文件的路径。

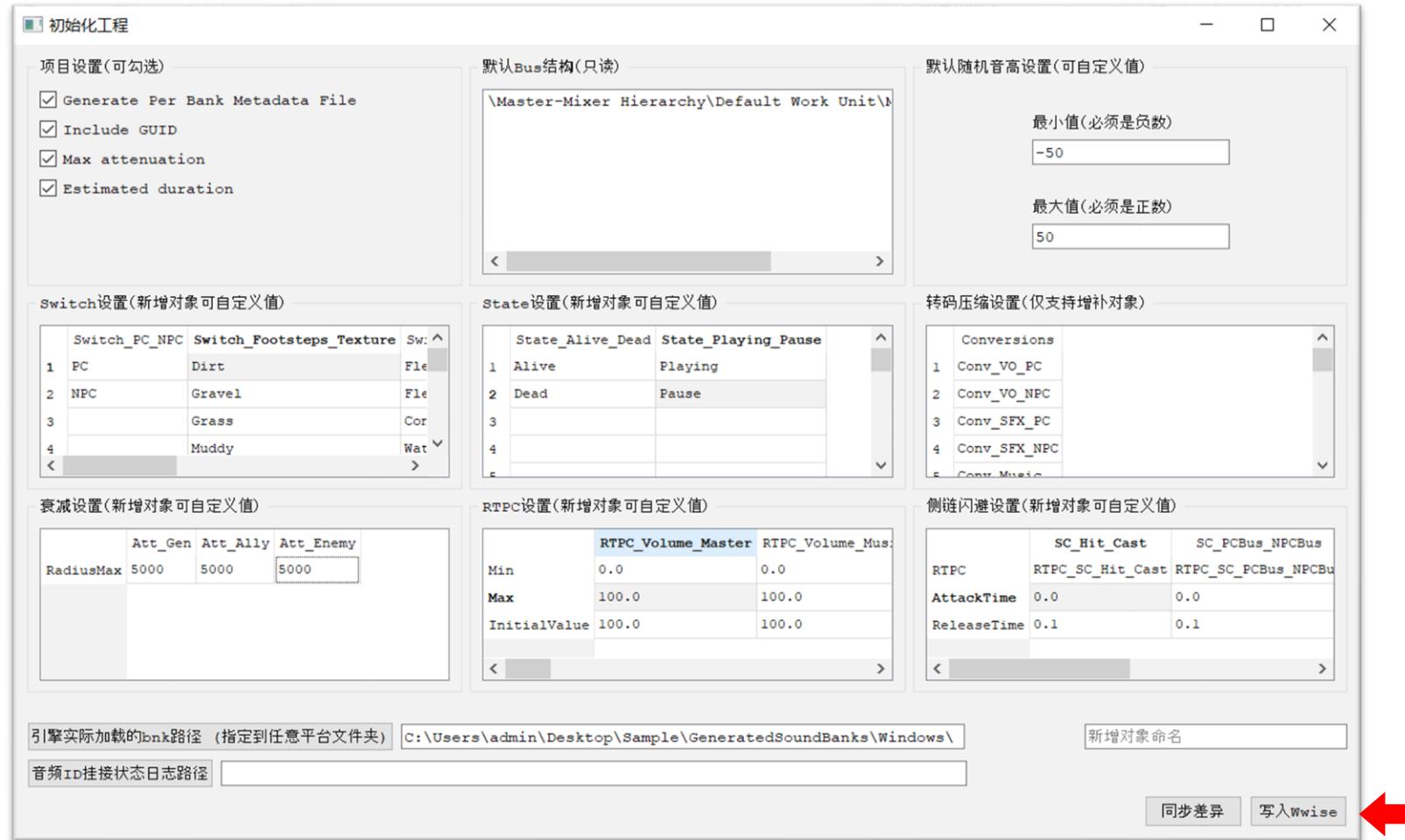
注：如果你的引擎环境中并不存在这样的功能，这个路径定位就没什么用，可以忽略。这里只是做一个解释说明，因为开发者自己的引擎环境里有^_^



初始化工程（首次）



编辑完所需的内容后，点击右下角的“写入Wwise”按钮。执行前确认窗口弹出，没问题的话，点击确认，即可将内容全部写入Wwise。



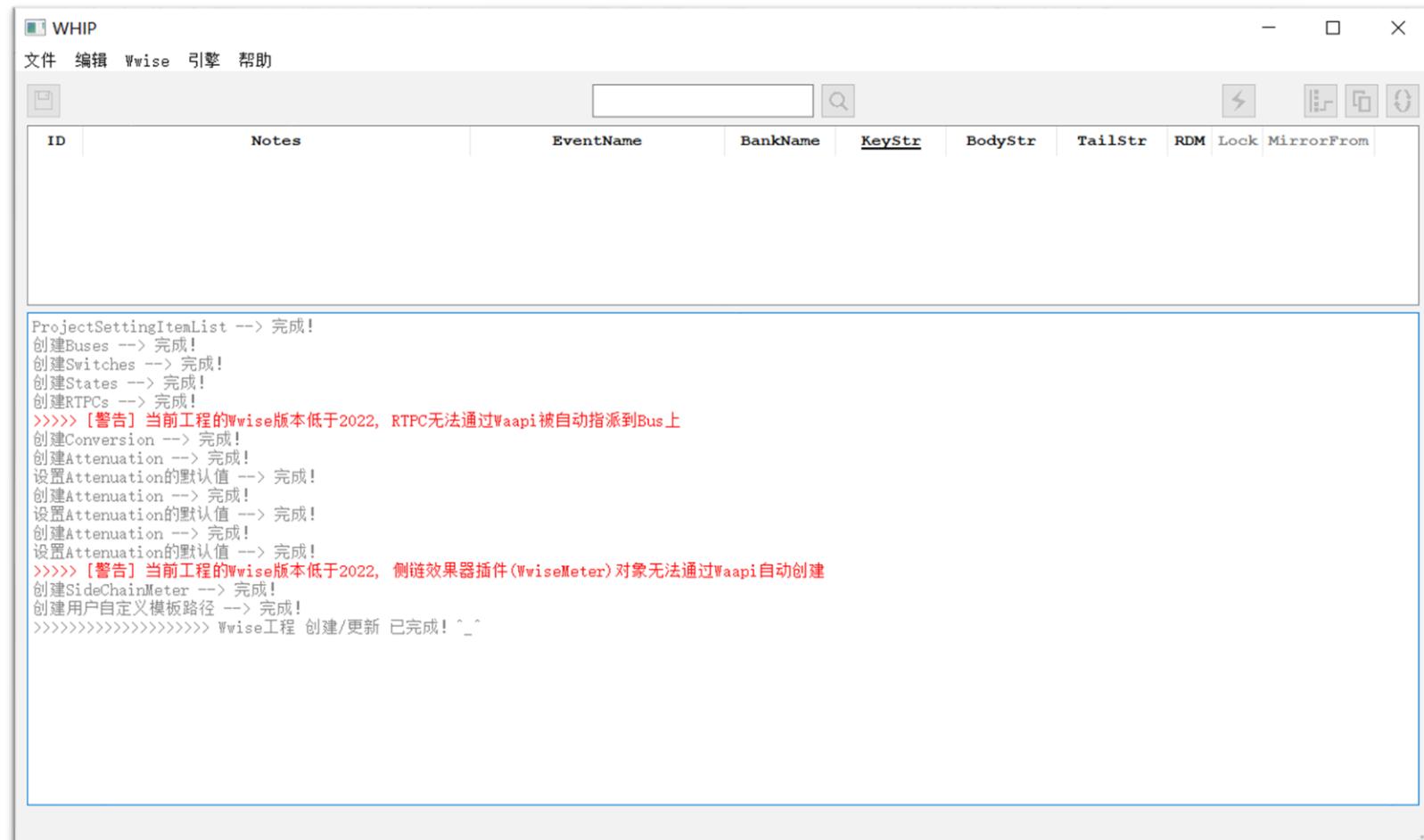
初始化工程（首次）

写入完毕后，初始化工程窗口将自动关闭。我们可以在主界面的日志区域中查看刚才的执行日志，然后在Wwise中检查结果。

注：由于一部分WA API接口是2022版本以上才提供的，有的内容低版本并不能正常写入，于是可能看到类似下图中的个别警告提示。当然，通过直接改写wwu也可以，只是暂时没这样处理。不过没关系，不支持自动化的部分，后续手动在Wwise里创建就好。

(图中第一行关于RPC的红字提示，指的是创建了Bus后，没能把几条分别控制总线、SFX、VO、Music输出音量的RPC，分别指派到各自的Bus上)

温馨提示：请尽量考虑使用尽可能高、且相对可靠的Wwise版本^_^

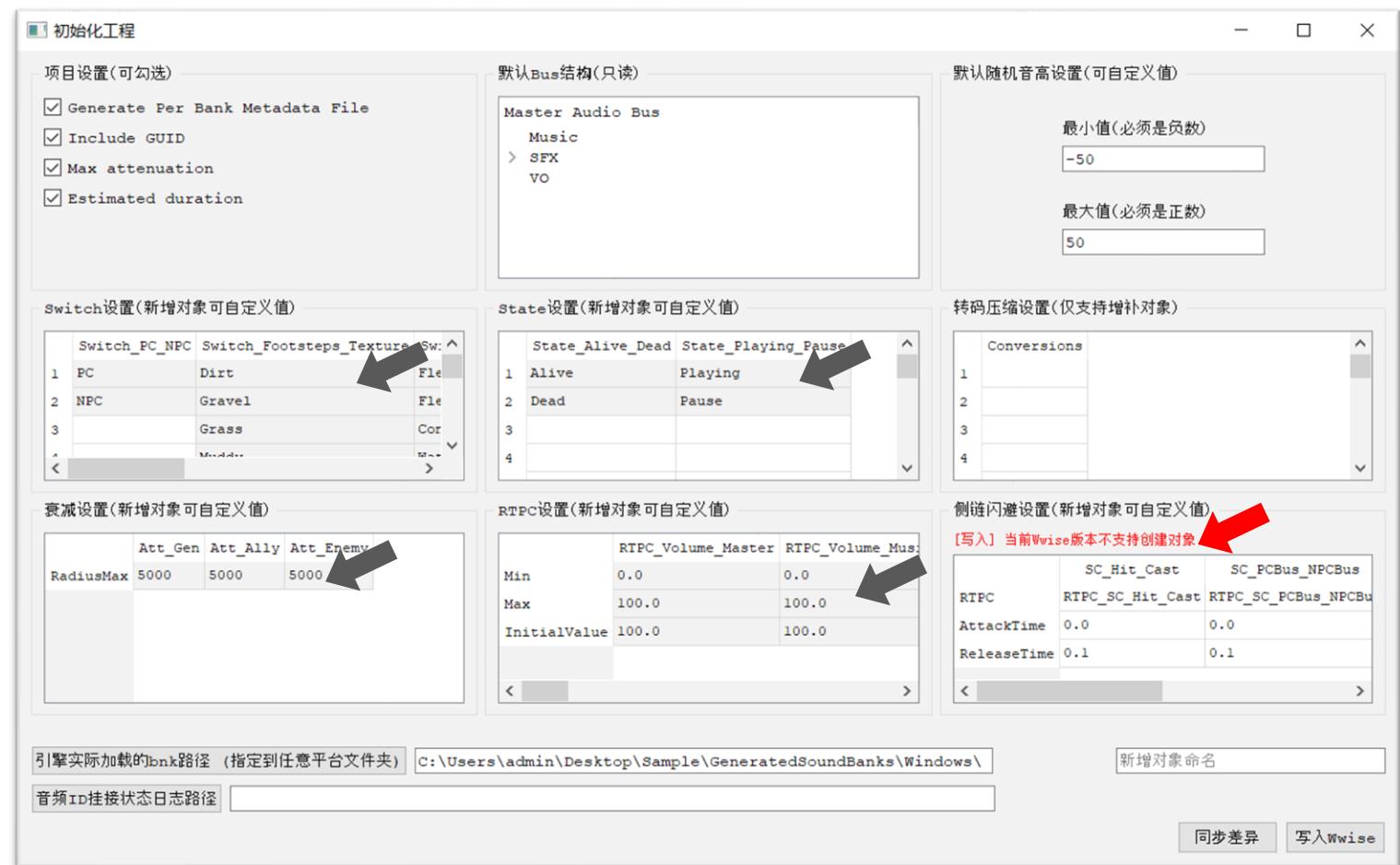


初始化工程（非首次）



如果Wwise工程并不是一个干净的新工程，且之前已经有过许多内容了，
我们打开初始化工程窗口，点击“同步差异”按钮后，会看到Wwise内已存在的内容显示出来、呈灰底不可编辑的状态。
安全起见，已存在的信息不允许通过插件二次修改，只能手动在Wwise里改。不过我们可以继续通过插件窗口新增Wwise里没有的内容。

注：另外也能看到一些功能不支持的重新提示。
读取或写入不支持的，不会生效，就不必折腾了。



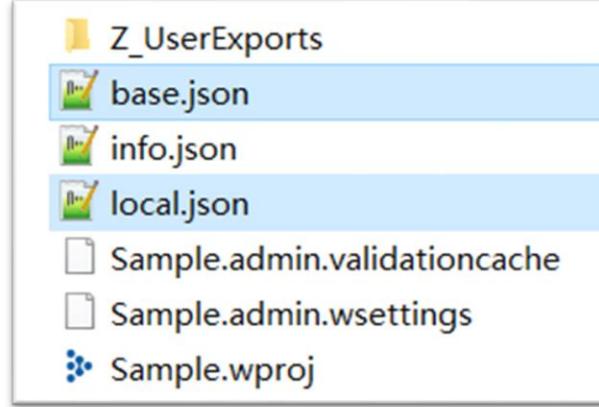
初始化工程（非首次）



温馨提示：

初始化工程窗口中的数据，一部分将被记录在当前Wwise工程目录下的base.json文件中，另一部分将被记录在local.json文件中。
初始化工程窗口中的文本变更，会在输入完成后自动保存，无需手动保存。

注：为了数据安全和插件运行的稳定性，请通过插件窗口来修改内容，尽量不要直接手动更改JSON文本！



local.json

```
"$ProjectStr$": "Sample.wproj",
-----[Local Path Info]-----:
"ActualGeneratedSoundBankPathOfOnePlatform": "C:\\\\Users\\\\admin\\\\Desktop\\\\Sam
"Path_SoundIDStatusReport": "C:\\\\Program Files (x86)\\\\Audiokinetic\\\\Wwise 20
"Path_DefaultSaveAsFolder": "C:\\\\Users\\\\admin\\\\Desktop"
```

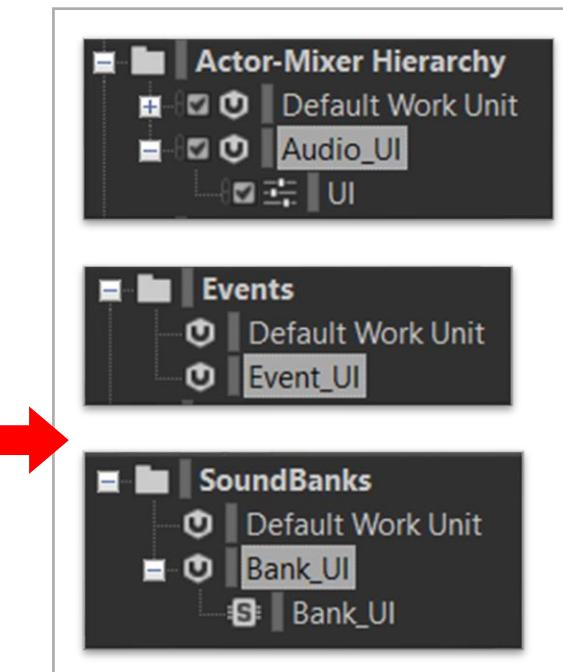
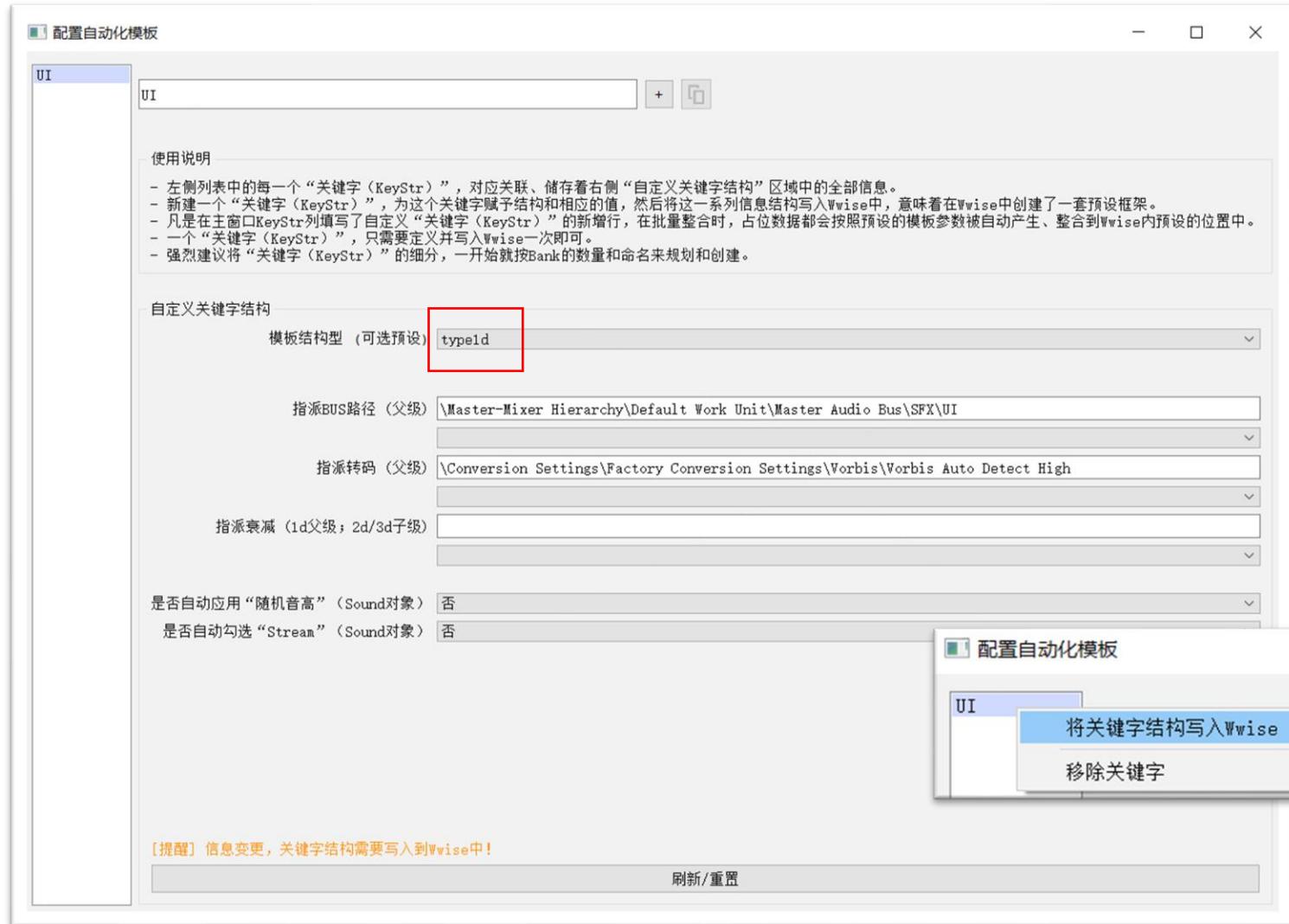
base.json

```
"Init_ProjectSettings": {
  "GenerateMultipleBanks": "True",
  "SoundBankGeneratePrintGUID": "True",
  "SoundBankGenerateMaxAttenuationInfo": "True",
  "SoundBankGenerateEstimatedDuration": "True"
},
"Init_BUS": {
  "Master Audio Bus": {
    "SFX": {
      "Action": {
        "PC": "",
        "NPC": ""
      },
      "Amb": "",
      "UI": ""
    },
    "Music": "",
    "VO": "",
    "CG": ""
  }
},
"Init_Switch": {
  "Switch_PC_NPC": [
    "PC",
    "NPC"
  ],
  "Switch_Footsteps_Texture": [
    "Dirt",
    "Gravel"
  ]
}
```

如何整合一个单纯的UI音效 (typeid)



第一步：通过“配置自动化模板”窗口，新增一个关键字，比如：UI。为关键字UI设置一系列属性后，右键关键字，将关键字结构自动写入Wwise。



如何整合一个单纯的UI音效 (typeld)



第二步：在主界面中，新增一行。将关键字UI填入KeyStr列，作为命名的开头。然后填写命名的其他部分、填写“RDM”随机数和“Notes”注释。

第三步：保存后，点击“一键整合”按钮。安全预检窗口弹出，提示通过后，点击确认，等待自动化整合完成。若提示不通过，先修复问题后，再重新执行。

A screenshot of a software interface showing a table with the following columns: ID, Notes, EventName, BankName, KeyStr, BodyStr, TailStr, RDM, and a 'One-click Integration' button. The 'KeyStr' column is highlighted in blue, and the 'One-click Integration' button has a large blue arrow pointing towards it.

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	One-click Integration
1	10001 UI, 装备升级提示			UI	Equip	LevelUp	1	



第四步：执行完毕后，关注表格中EventName、BankName、Lock列的变化。打开Wwise，确认整合结果。
保存表格数据，用正式资源覆盖临时[静音替条](#)，流程结束。

A screenshot of the software interface showing the same table after integration. The 'EventName' and 'BankName' columns are highlighted with red boxes. The 'EventName' cell contains 'Play_UI_Equip_LevelUp' and the 'BankName' cell contains 'Bank_UI'.

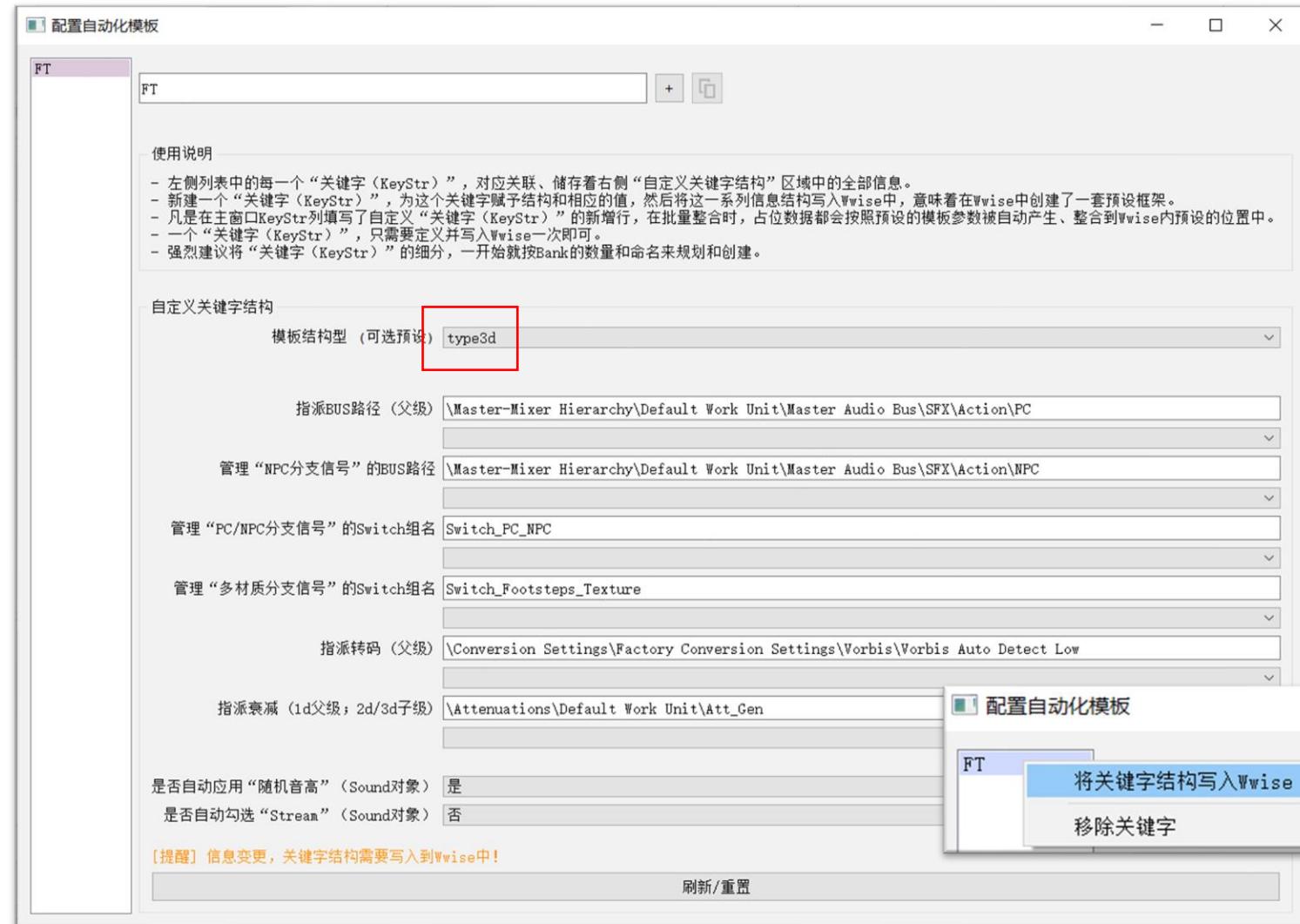
ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1	10001 UI, 装备升级提示	Play_UI_Equip_LevelUp	Bank_UI	UI	Equip	LevelUp	1	1	

如何整合一个单纯的UI音效 (type1d)

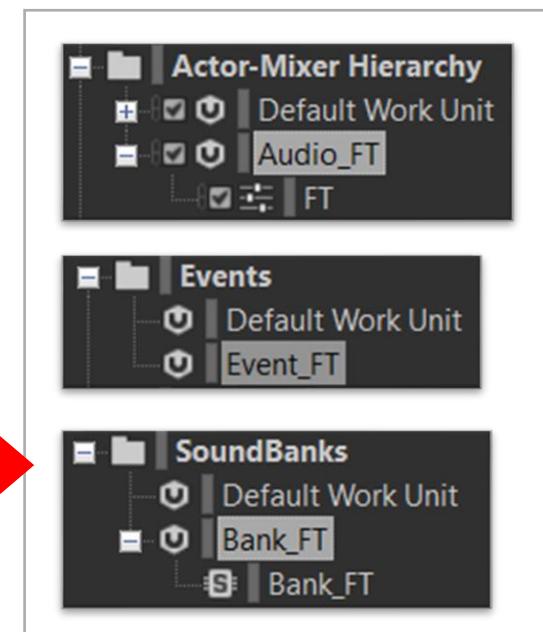
查看结果：

如何整合一套玩家的脚步声 (type3d)

第一步：通过“配置自动化模板”窗口，新增一个关键字，比如：FT。为关键字FT设置一系列属性后，右键关键字，将关键字结构自动写入Wwise。



关于新建关键字及属性，详细的操作说明，请阅读“[一劳永逸！配置自动化模板！](#)”章节



如何整合一套玩家的脚步声 (type3d)



第二步：在主界面中，新增一行。将关键字FT填入KeyStr列，作为命名的开头。然后填写命名的其他部分、填写“RDM”随机数和“Notes”注释。

第三步：保存后，点击“一键整合”按钮。安全预检窗口弹出，提示通过后，点击确认，等待自动化整合完成。若提示不通过，先修复问题后，再重新执行。

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	One Key Integration	MirrorFrom
1	10001 角色通用脚步			FT	Gen		4		



第四步：执行完毕后，关注表格中EventName、BankName、Lock列变化。打开Wwise，确认整合结果。
保存表格数据，用正式资源覆盖临时静音替条，流程结束。

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1	10001 角色通用脚步	Play_FT_Gen	Bank_FT	FT	Gen		4	1	

如何整合一套玩家的脚步声 (type3d)



查看结果：

The screenshot displays the Wwise Editor interface with three main windows:

- Wwise Editor Main Window:** Shows a table with one row of data:

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001	角色通用脚步	Play_FT_Gen	Bank_FT	FT	Gen		4	1	

A large text area below the table contains log messages indicating the successful import of various sound files into RandomContainers.
- Actor-Mixer Hierarchy:** A tree view showing the audio setup:
 - Default Work Unit
 - Audio_FT
 - FT
 - FT_Gen
 - Dirt
 - FT_Gen_Dirt_01
 - FT_Gen_Dirt_02
 - FT_Gen_Dirt_03
 - FT_Gen_Dirt_04
 - Grass
 - Gravel
 - Metal
 - Muddy
 - Snow
 - Stone
 - Water_Ankle
 - Water_Knee
 - Water_Puddle
 - Wood
 - PC
 - Dirt
- Event Property Editor:** Shows the properties for the event "Play_FT_Gen":

Icon	Inclusion	Type	Target
▶	<input checked="" type="checkbox"/>	Play	FT_Gen

如何整合一套玩家的脚步声 (type3d)



【补充说明】

对于通过type3d（包括type2d、type2d_vo）生成的数据，PC和NPC两边容器里的wav资源是复用了同一份，资源层面上并没有增倍。并不是PC、NPC各自独立的两套，而是同一套wav资源被分别指派到了PC和NPC两边。

然而，有些时候，设计师不希望PC和NPC共用同一套资源，而是要有区别，那么typet（用户自定义类）的模板类型，会比type3d的模板更为适合。typet的具体配置方式，可以参考[typet示例](#)。

*扩展提示：

在type1d和type3d之间，还有一个type2d的模板类型。

type2d的属性结构，相比起type3d，仅保留了适应于多人信号切换的Switch_PC_NPC层级，减少了适应于材质切换的Texture层级。

因此，若设计师想要创建一个不涉及材质切换的声音，比如，玩家拿着火把空挥的Whoosh，则可在配置关键字属性的时候，选择type2d。

*温馨提示：

type3d模板结构，适用于涉及多角色类型、材质切换的资源的整合包装。

比如，PC和NPC信号需要各自独立管理、脚步材质涉及切换的基础移动音效；或涉及武器命中不同材质、攻击方信号通路需要区别管理的碰撞音效。

type3d是一个适用于管理“PC/NPC信号差异 + 其他Switch切换”的“双层Switch维度”的模板类型。

同理，type2d是一个适用于管理“PC/NPC信号差异”的“单层Switch维度”模板类型。

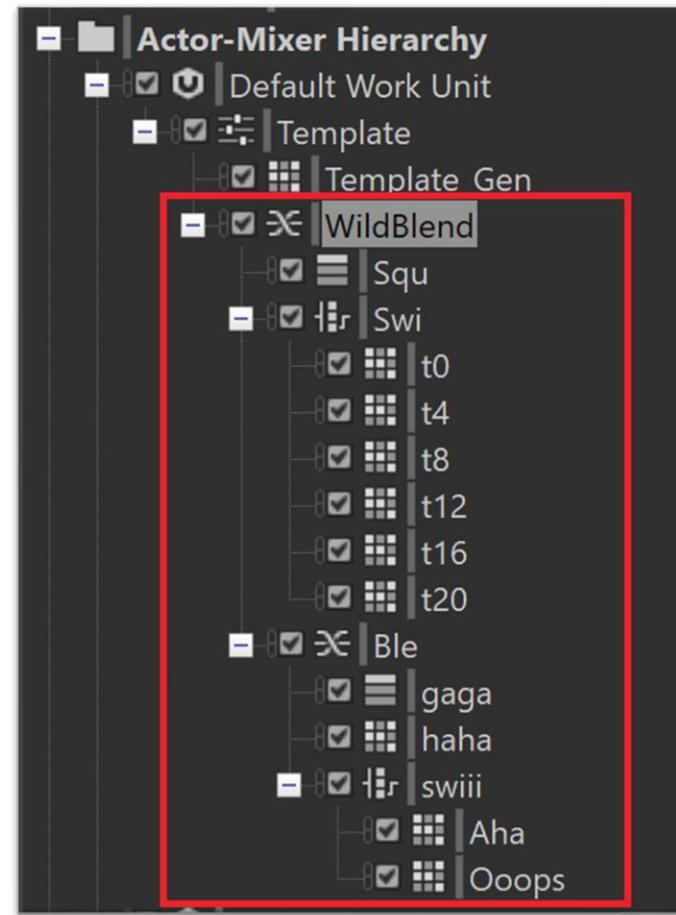
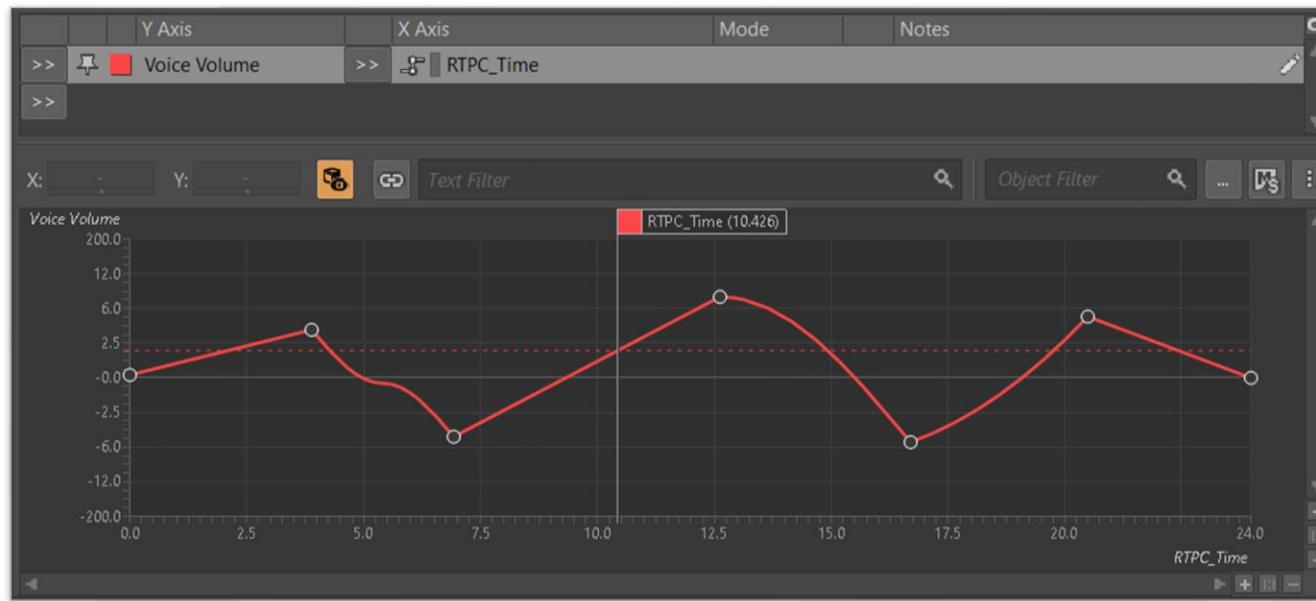
我有一个自定义的模板结构，应用广泛，但结构比较特殊， 插件的三个预置基础模板框架都无法应对，怎么办？（typet）



首先，设计师需要在WWISE工程内，手动自定义模板结构形态，比如，像图例这样。
为了更好地解释说明，示例模板结构故意做得比较复杂，以能够清晰展示整合后的结果细节。

*友情提示：

图例中的“WildBlend”对象纯粹是示例（是假定设计师正巧设计了这样一个形态），没有任何现实指导意义。
为了您和团队的身心健康，请尽量避免在真实工作中制造如此“热情奔放”的模板结构。

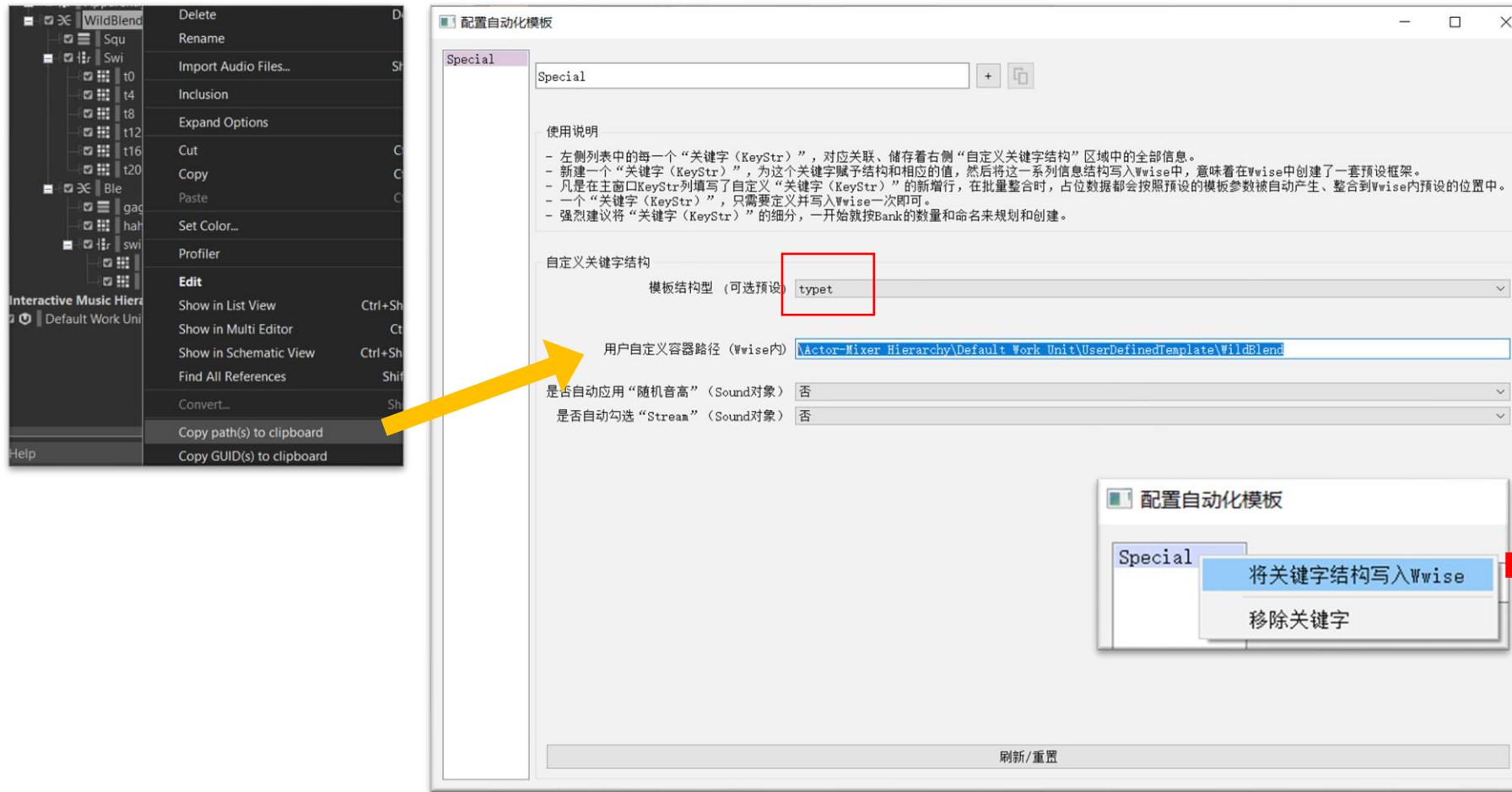


我有一个自定义的模板结构，应用广泛，但结构比较特殊， 插件的三个预置基础模板框架都无法应对，怎么办？（typet）

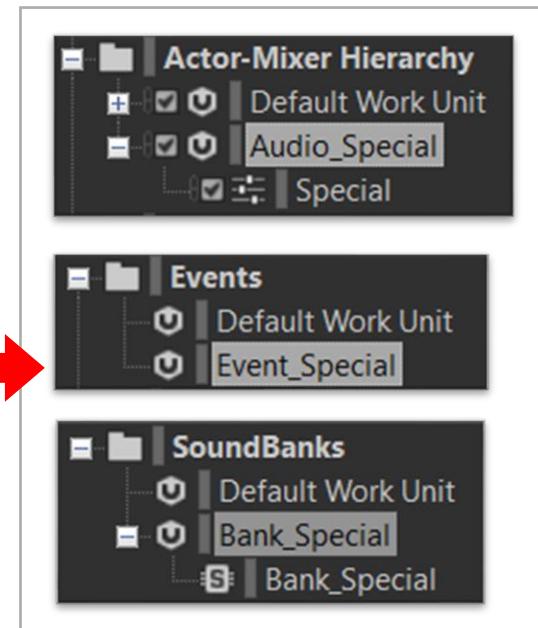


第一步：通过“配置自动化模板”窗口，新增一个关键字，比如：Special。

第二步：将Wwise工程中的模板路径复制，粘贴到“用户自定义容器路径中”。其他都确认无误后，右键关键字，将关键字结构自动写入Wwise。



关于新建关键字及属性，详细的操作说明，请阅读“[一劳永逸！配置自动化模板！](#)”章节



我有一个自定义的模板结构，应用广泛，但结构比较特殊， 插件的三个预置基础模板框架都无法应对，怎么办？（typet）

第三步：在主界面中，新增一行。将关键字Special填入KeyStr列，作为命名的开头。然后填写命名的其他部分、填写“RDM”随机数和“Notes”注释。
第四步：保存后，点击“一键整合”按钮。安全预检窗口弹出，提示通过后，点击确认，等待自动化整合完成。若提示不通过，先修复问题后，再重新执行。

The screenshot shows the typet application's main interface. At the top, there is a toolbar with various icons. Below it is a table with columns: ID, Notes, EventName, BankName, KeyStr, BodyStr, TailStr, RDM, and a button labeled '一键整合' (One-click Integration). A blue arrow points to the '一键整合' button. In the table, the 'KeyStr' column for the first row contains the value 'Special'. A second table below shows a single row with the same columns, but the 'EventName' column now contains 'Play_Special_Amazing'. A small window titled '安全预检通过' (Safety Check Passed) is displayed, containing a warning icon and the message '“一键整合”安全预检通过。确认开始执行?' (One-click integration safety check passed. Confirm to start execution?). It has '确认' (Confirm) and '取消' (Cancel) buttons.

第五步：执行完毕后，关注表格中EventName、BankName、Lock列变化。打开Wwise，确认整合结果。
保存表格数据，用正式资源覆盖临时[静音替条](#)，流程结束。

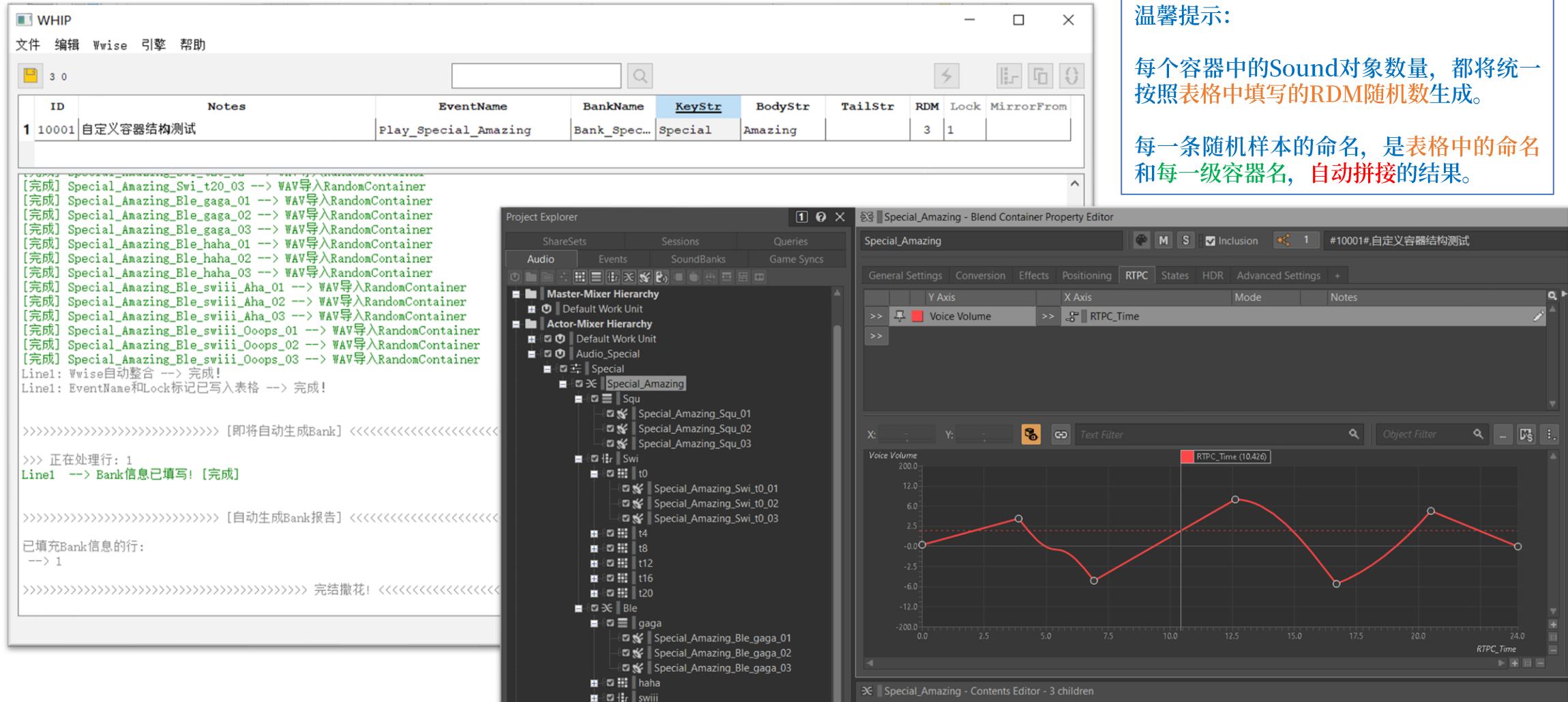
This screenshot shows the typet application's main interface again. The table now displays the updated data. The 'EventName' column for the first row is now 'Play_Special_Amazing'. The 'BankName' column is partially visible as 'Bank_Spec...'. The 'KeyStr' column still contains 'Special'. The 'BodyStr' column contains 'Amazing'. The 'TailStr' column is empty. The 'RDM' column contains '3'. The 'Lock' column contains '1'. The 'MirrorFrom' column is empty. The toolbar at the top includes a search bar and several icons.

温馨提示：

typet生成的数据，与type2d、3d不同的地方在于，每一条wav都是独立的资源，资源层面上并不存在复用。

我有一个自定义的模板结构，应用广泛，但结构比较特殊，插件的三个预置基础模板框架都无法应对，怎么办？(typet)

查看结果：





我有一个自定义的模板结构，应用广泛，但结构比较特殊， 插件的三个预置基础模板框架都无法应对，怎么办？（typet）

【补充说明】

对于插件2.0版本，出于一键批量自动化**数据安全性、易维护、易扩展**的考虑，开发者制定了如下规则：

- 1) 由于插件2.0版本尚未支持动态语音和互动音乐模块，因此，以下几个容器类型暂时列入了**禁用的范围**：
Voice多语言语音容器类型；**Music Switch Container**类型；**Music Playlist Container**类型；**Music Segment**类型
- 2) 允许自由组合使用的容器类型有：**Random Container**类型；**Sequence Container**类型；**Blend Container**类型；**Switch Container**类型。
禁用的容器类型有：**Sound**类型；**Actor-Mixer**类型；**Virtual Folder**文件夹类型。

*由于这三个类型的参与很容易给自动化以及后续维护带来许多不必要的麻烦，2.0版本暂时禁用了这些类型，待开发者找到适当的解决方案后，后续的版本会酌情将它们逐一召回。

- 3) 在合法的自定义容器嵌套结构中，最末一级的容器，不允许以一个孤零零的**Switch Container**容器作为结尾。

*这个单纯是为了提醒设计师，在做模板的时候，别遗漏了给**Switch**容器安排相应的内容指派。

不过，更前级的**Switch**容器暂时没有安排这样的安全检查，设计师忘了就真的忘了，自动化后的**Switch**容器中会自然产生一些孤零零的**Sound**。

- 4) 允许、但不建议存在“**连续三级都是Switch Container**”的结构。

*注：“一键扩展**Switch**”的功能，出于数据安全性、复杂性的考虑，暂不支持处理这种结构

关于命名规范



全局命名字规范：除英文大小写字母、数字和下划线之外，其他符号均为命名中不允许出现的非法字符！

首先，当我们想要在主界面表格中新增一个对象时，需要先在主界面中，添加一行。可以通过在菜单栏点选“编辑→添加行”添加，也可以通过快捷键添加。

然后，新增的行中，有几个单元格的内容，需要格外关注：

1) 在ID单元格输入一个ID值 **(必填项。每新增一行时，程序会自动顺次填充一个，通常不需要修改，除非有特殊需求)**

提示：ID是必填项！可以是数字，可以是字母，也可以是数字和字母的组合。一个ID对应一行数据，每个ID必须全局唯一，不允许重复。

2) 在KeyStr单元格中输入的值 **(必填项。必须是通过“配置自动化模板”窗口成功创建、并成功写入Wwise的关键字)**

提示：除了设计师创建过的字符串之外，其他字符串都将视为非法字符

3) 在BodyStr和TailStr单元格中，填写需要的命名字段。

提示：BodyStr或TailStr单元格，可以都填，也可以选填。如果两个单元格里都有信息，则最终命名会通过“下划线”拼接。

4) 在RDM单元格中，输入需要的随机数 **(必填项)**

提示：随机数的合法范围是 整数1-8。

5) 在Notes单元格中，输入需要的备注信息 **(不受“全局命名字规范”影响)**

提示：备注信息将在一键整合后，自动同步到WWISE工程相应的Event的Notes中。

6) EventName和BankName两个单元格的内容，将在自动整合后，通过插件功能自动填写，无需手动填写

7) 经历过自动化整合的行，都会在Lock单元格中自动写入一个字符，比如：“1”，以作为“已执行标记”。

如果用户在处理其他行时不小心带到“有执行标记的行”，则会在执行过程中自动跳过该行，并在日志中有所提示。

注：若后续需要通过自动化程序二次处理该行的数据，则需要先将Lock和BankName两个单元格都清空。

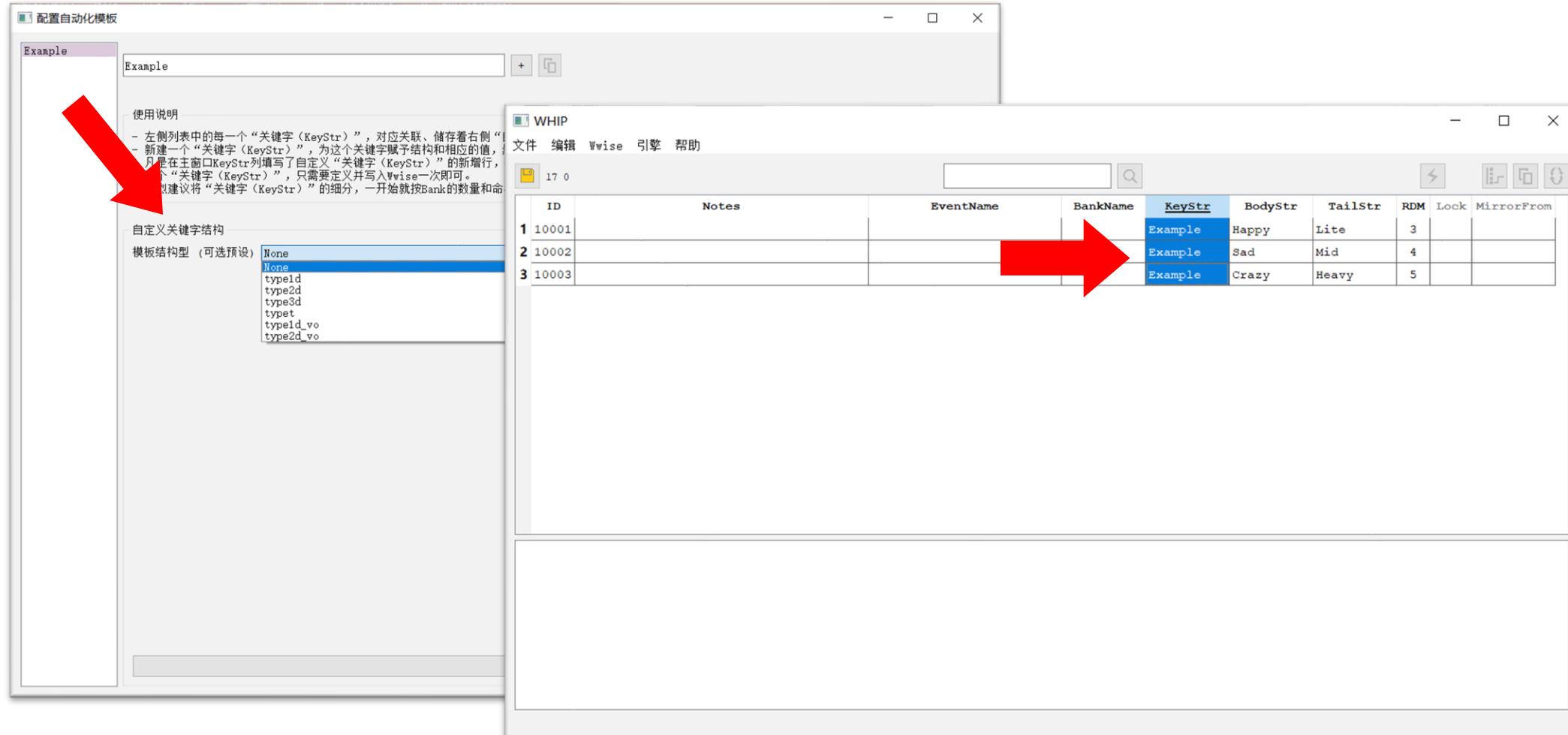
一劳永逸！配置自动化模板！



实现“命名即所有”的过程中，有一个环节尤为重要，那就是通过一个**关键字**作为桥梁，把**包含关键字的对象名**和**预设属性**关联起来。

原理：当我们新建一个关键字“Example”，赋予它一系列自定义属性，将属性框架写入Wwise后，

凡是在命名开头带有“Example”关键字的，在一键执行时，都将依照它“**自定义关键字结构**”中的属性设置，在Wwise中相应的WorkUnit中生成数据。

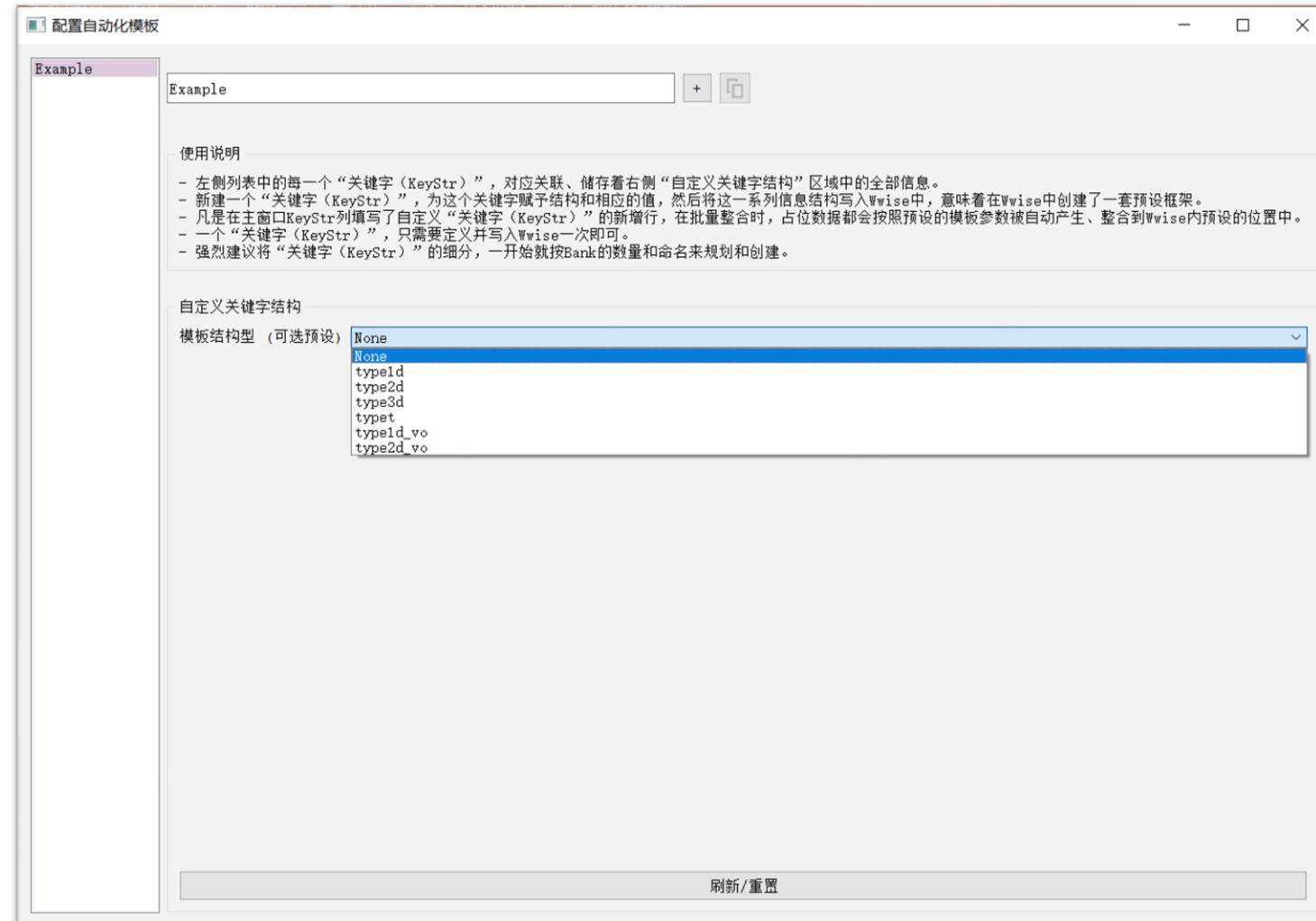


一劳永逸！配置自动化模板！



首先，在文本框中输入一个[关键字](#)，点击右边的“+”按钮。这时，新的关键字将出现在左侧的列表中，底色呈[粉红色](#)。

第一个需要为关键字设置的属性是“[模板结构形（可选预设）](#)”，不同的选择，也对应着不同的[容器结构形态](#)。



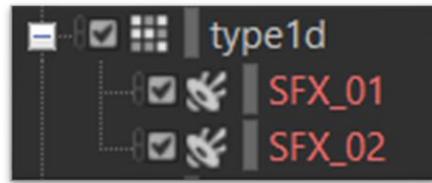
一劳永逸！配置自动化模板！



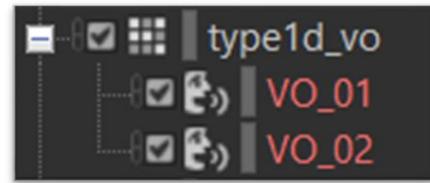
示例：以下是每一种模板结构型，对应的不同的容器结构形态。

除type1可任由设计师自定义之外，其他容器结构都是固定的、即时可用的形态。

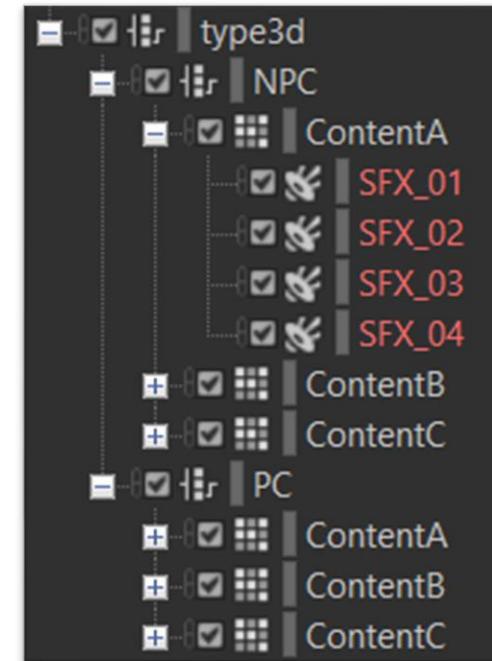
[type1d](#): 单一随机容器



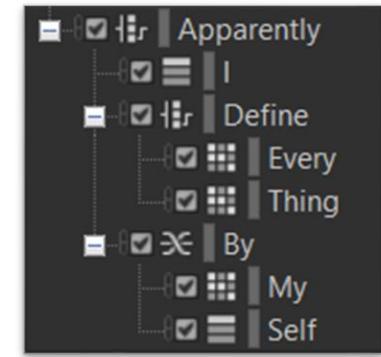
[type1d_vo](#): 单一随机容器（语音）



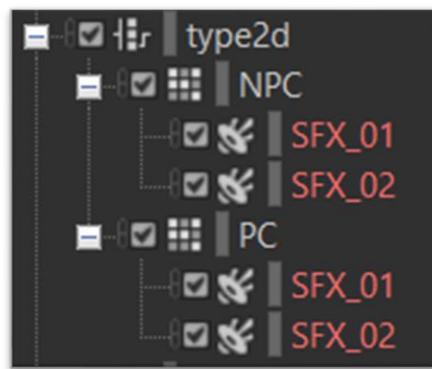
[type3d](#): 双层Switch容器



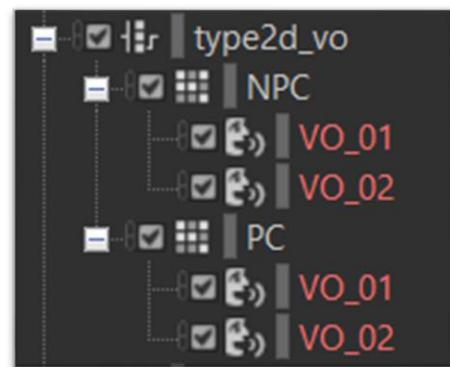
[type1](#): 自定义容器



[type2d](#): 单层Switch容器



[type2d_vo](#): 单层Switch容器（语音）



注：以上几种结构型，可能大致能够覆盖到绝大多数游戏类型、大部分对象的实用需要。

一劳永逸！配置自动化模板！



示例： [type1d](#)、[type1d vo](#)的属性

指派BUS路径（父级）：这里选择的是，通过当前关键字生成的最父级Actor-Mixer上的Bus。这样，所有的子对象的Bus都将默认随父级。

指派转码（父级）：同上，也是给最父级Actor-Mixer上选择指派一个Conversion

指派衰减：因为当前的模板结构型是1d的单一Random容器类型，因此，可以选择一个Attenuation对象直接指派到最父级Actor-Mixer上

是否自动应用“随机音高”：如果选“否”，则批量产生的Sound对象不会应用“初始化工程”界面中自定义好的[随机属性值](#)，否则将会自动应用

是否自定勾选“Stream”：如果选“否”，则批量产生的Sound对象不会自动勾选“Stream”，否则会自动勾选。

自定义关键字结构

模板结构型 (可选预设)	type1d
指派BUS路径 (父级)	
指派转码 (父级)	
指派衰减 (1d父级； 2d/3d子级)	
是否自动应用“随机音高” (Sound对象)	否
是否自动勾选“Stream” (Sound对象)	否

一劳永逸！配置自动化模板！



示例： [type2d](#)、[type2d vo](#)的属性中有两个与1d类型不一样的

管理“NPC分支信号”的BUS路径：这里选择的是，给每一个新增Switch容器下属中，对应NPC数据的随机容器上，指派一个管理NPC信号的BUS。

管理“PC/NPC分支信号的Switch组名”：给每一个新增Switch容器上，指派负责管理PC/NPC的SwitchGroup

指派衰减：由于当前是2d的类型，因此，Attenuation将会被指派到Switch容器下属中对应NPC数据的随机容器上，而不是最父级Actor-Mixer容器

自定义关键字结构

模板结构型（可选预设） [type2d](#)

指派BUS路径（父级）

管理“NPC分支信号”的BUS路径

管理“PC/NPC分支信号”的Switch组名

指派转码（父级）

指派衰减（1d父级；2d/3d子级）

是否自动应用“随机音高”（Sound对象） 否

是否自动勾选“Stream”（Sound对象） 否

The screenshot shows a configuration dialog box with several input fields. At the top, there's a dropdown menu set to 'type2d'. Below it are two main sections highlighted with red boxes: 'Management of NPC branch signals' and 'Attenuation'. The 'Management of NPC branch signals' section contains a dropdown menu for 'Assign BUS Path (Parent)'. The 'Attenuation' section also has a dropdown menu for 'Assign Attenuation (1d Parent; 2d/3d Child)'. At the bottom, there are two more sections: 'Automatically apply "Random Pitch" (Sound object)' (set to '否') and 'Automatically check "Stream" (Sound object)' (set to '否').

一劳永逸！配置自动化模板！



示例： [type3d](#) 的必要属性中又有一个与1d、2d类型不一样的

管理“多材质分支信号”的BUS路径：这里选择的是，给对应PC、NPC的Switch容器上，指派一个对应材质的SwitchGroup

自定义关键字结构

模板结构型（可选预设） [type3d](#)

指派BUS路径（父级）

管理“NPC分支信号”的BUS路径

管理“PC/NPC分支信号”的Switch组名

管理“多材质分支信号”的Switch组名

指派转码（父级）

指派衰减（1d父级；2d/3d子级）

是否自动应用“随机音高”（Sound对象） 否

是否自动勾选“Stream”（Sound对象） 否

一劳永逸！配置自动化模板！

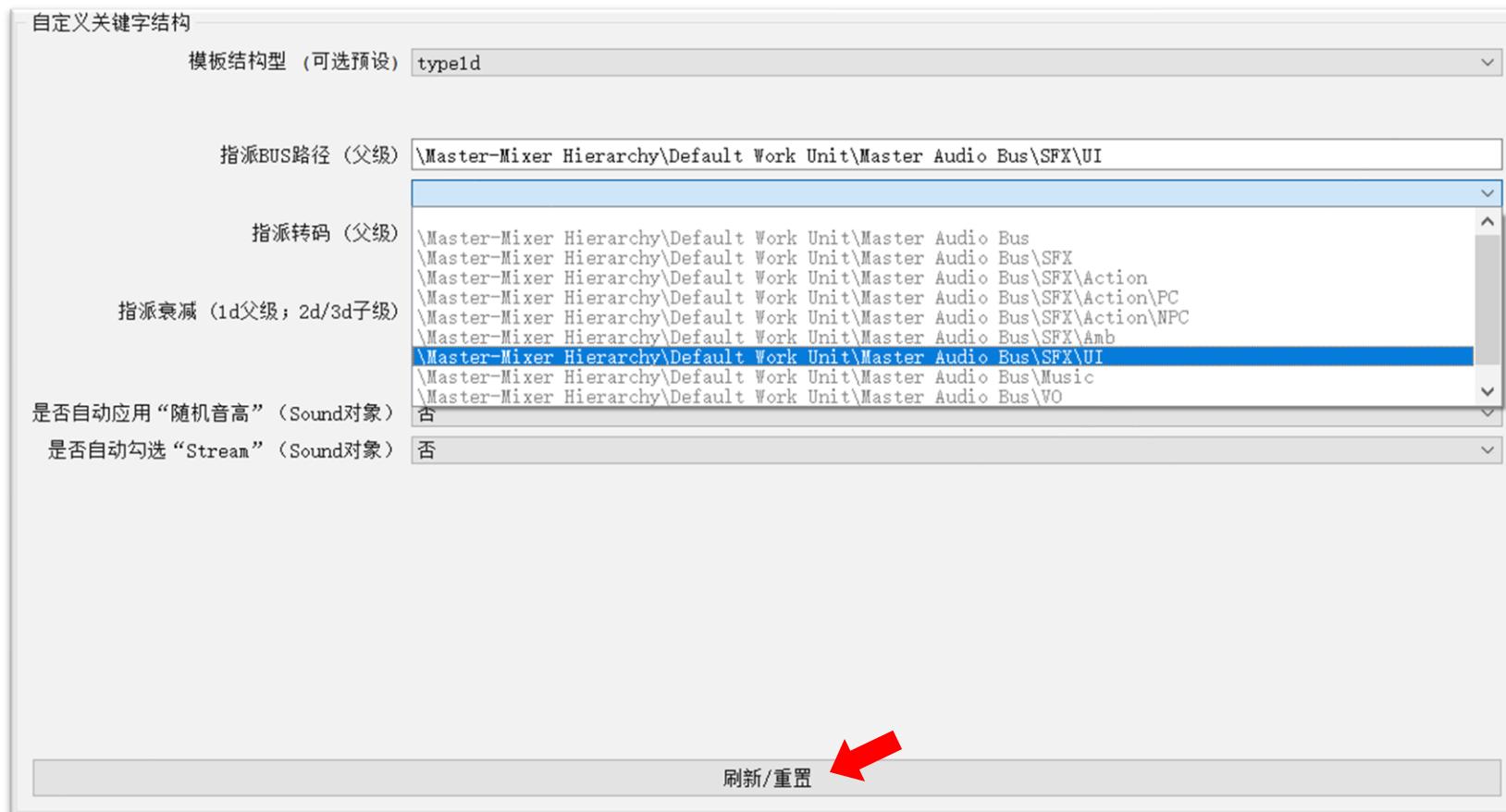


填写示例：type1d、type1d_vo、type2d、type2d_vo、type3d的属性

每一个文本框下方，都有一个下拉列表，显示着当前工程里现有的合法对象路径。

点选后，路径或对象名信息将直接显示在文本框中，无需手动填写。当然，也可以直接手动填写或修改。

如果下拉列表中的内容和Wwise里实际的内容不一致，则需要先点击下方的“刷新/重置”按钮，刷新下拉列表，再进行选择。

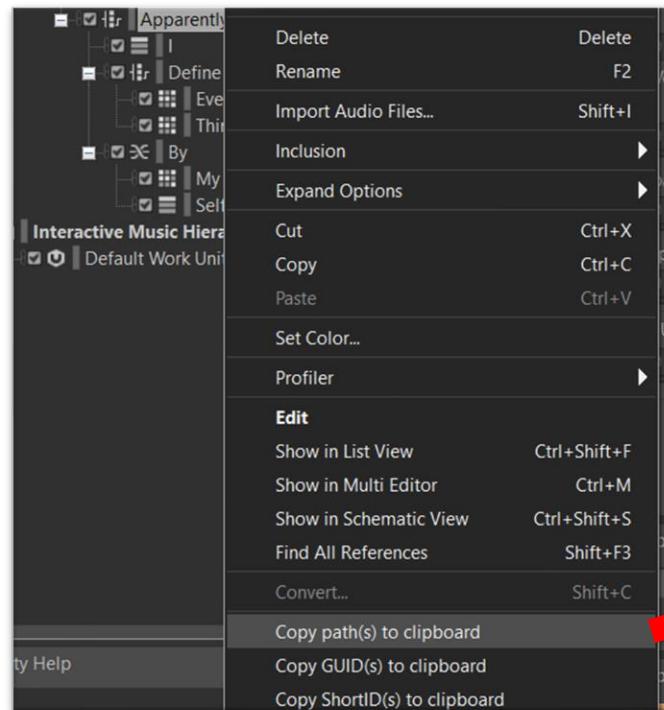


一劳永逸！配置自动化模板！



填写示例： [typet](#) 的属性比较少，其中有一个与其他类型都不一样的

用户自定义容器路径（Wwise内）：这里填写的是，用户在Wwise中自定义创建的模板路径。这个路径需要从Wwise中复制粘贴过来。



复制路径的方法：

- 1) 按住键盘上的Shift键
- 2) 鼠标右键点击Wwise中的目标容器对象
- 3) 选择Copy path(s) to clipboard
- 4) 松开Shift键。粘贴路径。



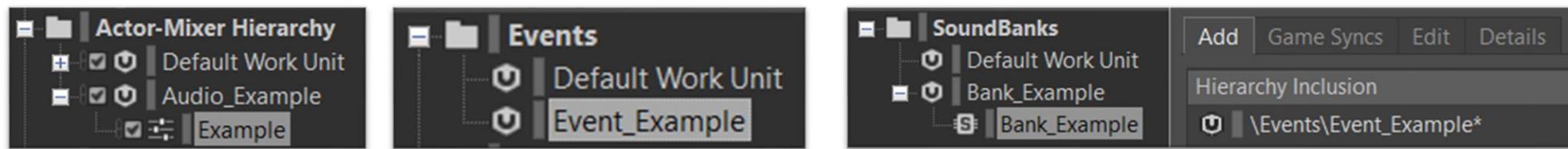
一劳永逸！配置自动化模板！



设置完属性后，对着左侧列表中新建的关键字，右键点选“将关键字结构写入Wwise”，等待看到窗口下方的写入提示。



返回Wwise，确认检查写入结果。如果没什么问题的话，流程结束。



温馨提示：

我们可以看到，每一个关键字，都对应着3个WorkUnit（Actor-Mixer、Events、SoundBanks），彼此呼应，整整齐齐。同时，每一个关键字，实际上也对应着一个bnk文件。因此，根据SoundBank的细分来规划关键字，是一个非常推荐的做法！

关键字的颗粒度，最好与bnk文件的颗粒度保持同步，越细致越好，尽量避免随心所欲的分类和命名。

一劳永逸！配置自动化模板！



当然，如果每新建一个关键字，都要从头到尾折腾一通，就很麻烦了。因此，在“+”按钮右边，还有一个[镜像](#)按钮。

例如：在左侧列表中选中要镜像的关键字Example，在新增的文本框中输入新的关键字New，然后点击[镜像](#)按钮。

这时候，新增的关键字New里的所有属性，会自动呈现为和Example一模一样的内容。如果没什么要调整的，可以直接右键写入Wwise。

至此，配置自动化模板的过程就已全部完成。
关键字可以参与主界面的命名了^_^

The screenshot shows two instances of the 'Configure Automation Template' dialog. The top instance has 'Example' selected in the list and 'New' entered in the text field. A blue arrow points from the '+' button area to the 'Mirror' button (indicated by a blue square with a white circle). The bottom instance shows the results of mirroring: 'New' is selected, and its properties (e.g., 'Template Structure Type: type3d', various bus paths, switch groups, and attenuation settings) are identical to those of 'Example'. A red arrow points from the bottom instance's list to the 'Custom KeyStr' section of the right panel, which is highlighted with a red border. A note at the bottom of the right panel says: '[提示] 镜像的字符串对象是 --> Example'.

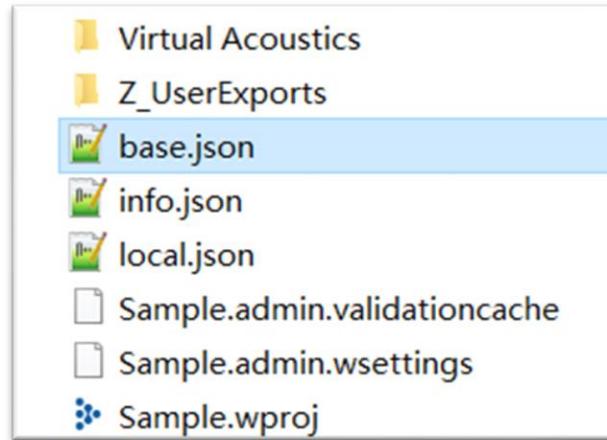
一劳永逸！配置自动化模板！



【补充提示】

配置自动化模板窗口中的所有数据，都将被记录在当前Wwise工程目录下的base.json文件中。该窗口中的文本变更会在检测到变更时自动保存。

注：为了数据安全和插件运行的稳定性，请通过插件窗口来修改内容，尽量不要手动直接更改JSON文本！



```
{
    "$ProjectStr$": "Sample.wproj",
    "-----[Key Info]-----": "-----",
    "Data_KeyInfo": {
        "Act": {
            "Structure_Type": "type2d",
            "Path_InWwise_UserDefinedTemplate": "",
            "Path_File_PlaceholderWAV": "cf\\wavPlaceholder\\silence.wav",
            "Path_Folder_TargetWAV": "Originals\\SFX\\",
            "Path_InWwise_TargetActorMixer": "\\Actor-Mixer Hierarchy\\Audio_Act\\Act",
            "Path_InWwise_TargetEvent": "\\Events\\Event_Act",
            "Path_InWwise_TargetBank": "\\SoundBanks\\Bank_Act\\Bank_Act",
            "Property_Conversion": "\\Conversion Settings\\Factory Conversion Settings\\Vorbis",
            "Property_Positioning": "\\Attenuations\\Default Work Unit\\Att_Ally",
            "Property_Bus": "\\Master-Mixer Hierarchy\\Default Work Unit\\Master Audio Bus\\S",
            "Property_Bus_NPC": "\\Master-Mixer Hierarchy\\Default Work Unit\\Master Audio Bu",
            "Property_SwitchGroupName_PC_NPC": "Switch_PC_NPC",
            "Property_SwitchGroupName_Texture": "",
            "Property_ifPitchRandom": "False",
            "Property_ifStream": "False"
        },
        "UI": {
            "Structure_Type": "type1d",
            "Path_InWwise_UserDefinedTemplate": "",
            "Path_File_PlaceholderWAV": "cf\\wavPlaceholder\\silence.wav",
            "Path_Folder_TargetWAV": "Originals\\SFX\\",
            "Path_InWwise_TargetActorMixer": "\\Actor-Mixer Hierarchy\\Audio_UI\\UI",
            "Path_InWwise_TargetEvent": "\\Events\\Event_UI",
            "Path_InWwise_TargetBank": "\\SoundBanks\\Bank_UI\\Bank_UI",
            "Property_Conversion": "",
            "Property_Positioning": ""
        }
    }
}
```

地面材质增加了！有许多涉及材质Switch的容器都需要扩充内容，怎么办？



假设：“Switch_Footsteps_Texture”的对象需要增加一个“Sand”，但影响面有6个之前已经整合过的对象。

The screenshot shows two windows side-by-side. On the left is a material switch editor titled "Switch_Footsteps_Texture". It lists various ground materials: Dirt, Grass, Gravel, Metal, Muddy, Snow, Stone, Water_Ankle, Water_Knee, Water_Puddle, and Wood. On the right is a software application titled "WHIP" which contains a table of footsteps events.

WHIP Table Data:

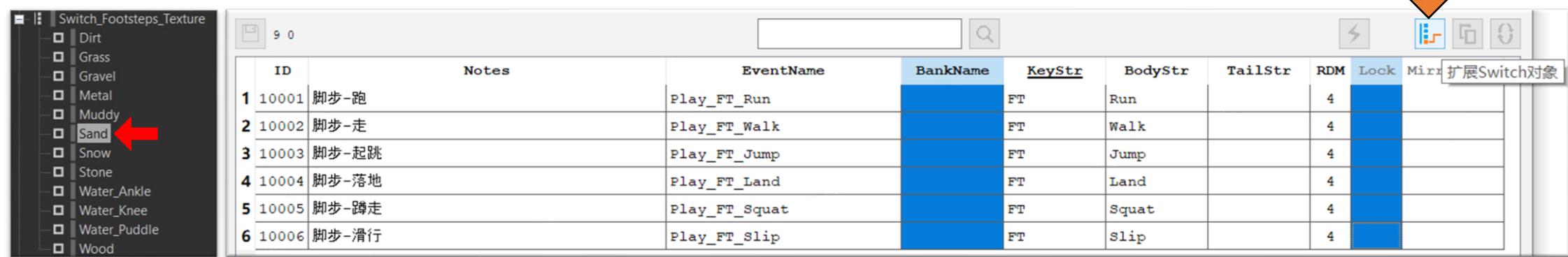
ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001	脚步-跑	Play_FT_Run	Bank_FT	FT	Run		4	1	
2 10002	脚步-走	Play_FT_Walk	Bank_FT	FT	Walk		4	1	
3 10003	脚步-起跳	Play_FT_Jump	Bank_FT	FT	Jump		4	1	
4 10004	脚步-落地	Play_FT_Land	Bank_FT	FT	Land		4	1	
5 10005	脚步-蹲走	Play_FT_Squat	Bank_FT	FT	Squat		4	1	
6 10006	脚步-滑行	Play_FT_Slip	Bank_FT	FT	Slip		4	1	

地面材质增加了！有许多涉及材质Switch的容器都需要扩充内容，怎么办？

第一步：在Wwise中，给“Switch_Footsteps_Texture”添加“Sand”对象，保存。

第二步：为表格中所有涉及增补Switch的行，**清空BankName列**的内容以辅助识别二次整合后的新内容，再**清空Lock列**的内容以**解锁**。保存表格。

第三步：选中需要处理的行（任意单元格即可），点击**“扩展Switch”**按钮。若安全预检通过，点击确认开始执行。若未通过，修复问题后重新点击。



ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001	脚步-跑	Play_FT_Run	FT	Run			4		
2 10002	脚步-走	Play_FT_Walk	FT	Walk			4		
3 10003	脚步-起跳	Play_FT_Jump	FT	Jump			4		
4 10004	脚步-落地	Play_FT_Land	FT	Land			4		
5 10005	脚步-蹲走	Play_FT_Squat	FT	Squat			4		
6 10006	脚步-滑行	Play_FT_Slip	FT	Slip			4		



第四步：执行完毕后，关注表格中EventName、BankName、Lock列变化。打开Wwise，确认整合结果。保存表格，用正式资源覆盖临时静音替条，流程结束。



ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001	脚步-跑	Play_FT_Run	Bank_FT	FT	Run		4	1	
2 10002	脚步-走	Play_FT_Walk	Bank_FT	FT	Walk		4	1	
3 10003	脚步-起跳	Play_FT_Jump	Bank_FT	FT	Jump		4	1	
4 10004	脚步-落地	Play_FT_Land	Bank_FT	FT	Land		4	1	
5 10005	脚步-蹲走	Play_FT_Squat	Bank_FT	FT	Squat		4	1	
6 10006	脚步-滑行	Play_FT_Slip	Bank_FT	FT	Slip		4	1	

地面材质增加了！有许多涉及材质Switch的容器都需要扩充内容，怎么办？

查看结果：

The screenshot shows the WHIP (Wwise Host Integration Platform) interface. On the left, there is a table of events:

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001	脚步-跑	Play_FT_Run	Bank_FT	FT	Run		4	1	
2 10002	脚步-走	Play_FT_Walk	Bank_FT	FT	Walk		4	1	
3 10003	脚步-起跳	Play_FT_Jump	Bank_FT	FT	Jump		4	1	
4 10004	脚步-落地	Play_FT_Land	Bank_FT	FT	Land		4	1	
5 10005	脚步-蹲走	Play_FT_Squat	Bank_FT	FT	Squat		4	1	
6 10006	脚步-滑行	Play_FT_Slip	Bank_FT	FT	Slip		4	1	

The right side shows a detailed view of the 'FT_Squat' audio object. It includes a tree view of sub-objects like 'FT_Jump', 'FT_Land', etc., and a parameter editor on the right. A red box highlights the 'Pitch' parameter, which is set to 0. Another red box highlights the 'FT_Squat_Sand_01' entry in the 'Sand' sub-object list.

```
Line5: 发现一个ObjectRef对象！ --> 已记录！
安全检查通过！继续下一步……
PC -----> 检测到对象差异！
Sand -----> 已创建！
Sand -----> 已指派关联！
新增容器均已生成，并均已完成指派！下一步将生成WAV……
NPC -----> 检测到对象差异！
Sand -----> 已创建！
Sand -----> 已指派关联！
新增容器均已生成，并均已完成指派！下一步将生成WAV……
Line5: 新的对象已生成，并已完成指派！
确认模板类型 -----> type3d
Line5: 获取到待执行的目标信息组！ --> 已记录！
[完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Squat_Sand_01.wav 已生成
[完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Squat_Sand_03.wav 已生成
[完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Squat_Sand_04.wav 已生成
[完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Squat_Sand_02.wav 已生成
Line5: WAV创建成功 --> 完成！
Line5: 正在生成Wwise工程内数据 --> 进行中……
FT_Squat_Sand_01 -----> WAV已导入
FT_Squat_Sand_02 -----> WAV已导入
FT_Squat_Sand_03 -----> WAV已导入
FT_Squat_Sand_04 -----> WAV已导入
FT_Squat_Sand_01 -----> WMA已导入
```

想完全复用一个Event，但从里到外的容器、资源命名，都得是全新的！怎么办？

假设：我们想复用“脚步-跑”的内容，把它们再镜像出一份“脚步-冲刺”的内容，但命名从头到脚都要换成新的。

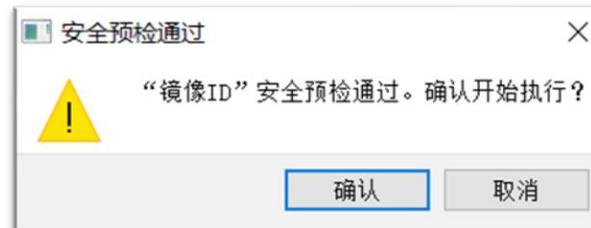
第一步：在表格中新增ID为10002的行，在关键字KeyStr列的单元格内，填写与ID为10001行一样的关键字；[RDM单元格内的数量可能也要一致](#)。

第二步：在新增行的MirrorFrom单元格中，填写镜像目标的ID：10001。保存表格。

第三步：选中需要处理的行（任意单元格即可），点击“[镜像ID](#)”按钮。若安全预检通过，点击确认开始执行。若未通过，修复问题后重试。



ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1	脚步-跑	Play_FT_Run	Bank_FT	FT	Run		4	1	
2	脚步-冲刺			FT	Sprint		4		10001



第四步：执行完毕后，关注表格中EventName、BankName、Lock列变化。打开Wwise，确认整合结果。保存表格，用正式资源覆盖[临时静音替条](#)，流程结束。

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1	脚步-跑	Play_FT_Run	Bank_FT	FT	Run		4	1	
2	脚步-冲刺	Play_FT_Sprint	Bank_FT	FT	Sprint		4	1	10001

想完全复用一个Event，但从里到外的容器、资源命名，都得是全新的！怎么办？

查看结果：

The screenshot shows the WHIP (Wwise Host Integration Platform) interface. At the top, there's a menu bar with File, Edit, Wwise, Engine, and Help. Below the menu is a toolbar with icons for search, copy, paste, and others. A table below the toolbar lists two events:

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1	脚步-跑	Play_FT_Run	Bank_FT	FT	Run		4	1	
2	脚步-冲刺	Play_FT_Sprint	Bank_FT	FT	Sprint		4	1	10001

Below the table, a message box displays the generation logs for each event:

- [完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_01.wav 已生成
C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Run_Sand_01.wav -----> C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_01.wav
- [完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_02.wav 已生成
C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Run_Sand_02.wav -----> C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_02.wav
- [完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_03.wav 已生成
C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Run_Sand_03.wav -----> C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_03.wav
- [完成] C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_04.wav 已生成
C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Run_Sand_04.wav -----> C:\Users\admin\Desktop\Sample\Originals\SFX\FT_Sprint_Sand_04.wav

At the bottom, there's a log window showing the process of creating audio assets:

```
Line2: WAV创建成功 --> 完成!  
Line2: 正在获取模板类型 --> 检索中  
Line2: 正在生成Wwise工程内数据 --> 进行中……  
[完成] PC --> 导入Switch对象，并设定指派  
[完成] NPC --> 导入Switch对象，并设定指派  
[完成] PC --> 给第二层级设定Switch指派  
[完成] NPC --> 给第二层级设定Switch指派  
[完成] Dirt --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Gravel --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Grass --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Muddy --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Wood --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Snow --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Stone --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Metal --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Water_Puddle --> 创建RandomContainer对象，并指派给相应的Switch  
[完成] Water_Ankle --> 创建RandomContainer对象，并指派给相应的Switch
```

温馨提示：

10002的内容，完全是基于10001镜像、然后修改命名后的结果。

新增对象的命名中，关键字必须与镜像目标一致。

对于type1d、type2d而言，随机数可以与镜像目标不一致，但type3d的随机数必须一致

The screenshot shows the Wwise Editor interface. On the left, the Event tree is displayed:

- Audio_FT
 - FT
 - FT_Jump
 - FT_Land
 - FT_Run
 - FT_Slip
 - FT_Sprint
 - NPC
 - Dirt
 - FT_Sprint_Dirt_01
 - FT_Sprint_Dirt_02
 - FT_Sprint_Dirt_03
 - FT_Sprint_Dirt_04

On the right, the parameters for the selected event, FT_Sprint_Dirt_01, are shown:

 - Volume: 0
 - Pitch: 0
 - Low-pass filter: 0
 - High-pass filter: 0

另起炉灶！AI上一轮生成的效果一般，再来一遍！（开发中，仅功能示意）

假设：我们刚刚通过AI一键整合生成过一批技能素材，实测后觉得差点意思，想在不新增对象、不改变命名的情况下，重新通过AI再生成一组新的。

第一步：为表格中所有涉及增补Switch的行，清空BankName列的内容以辅助识别二次整合后的新内容，再清空Lock列的内容以解锁。保存表格。

第二步：选中需要处理的行（任意单元格即可），点击“另起炉灶”按钮。若安全预检通过，点击确认开始执行。若未通过，修复问题后重新点击。

The screenshot shows a software interface with a table and a confirmation dialog.

Table:

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom	另起炉灶
1	10001 连击-释放	Play_Act_Combo_Cast		Act	Combo	Cast	3			
2	10002 连击-破坏	Play_Act_Combo_Smash		Act	Combo	Smash	3			
3	10003 连击-收尾	Play_Act_Combo_Tail		Act	Combo	Tail	3			

Confirmation Dialog:

安全预检通过

“另起炉灶”安全预检通过。确认开始执行？

确认 取消

An orange arrow points from the "Ren Qi Lu Zhao" button in the top right of the table to the "Confirm" button in the dialog box.

第三步：执行完毕后，关注表格中EventName、BankName、Lock列变化。打开Wwise，确认整合结果。保存表格数据，流程结束。

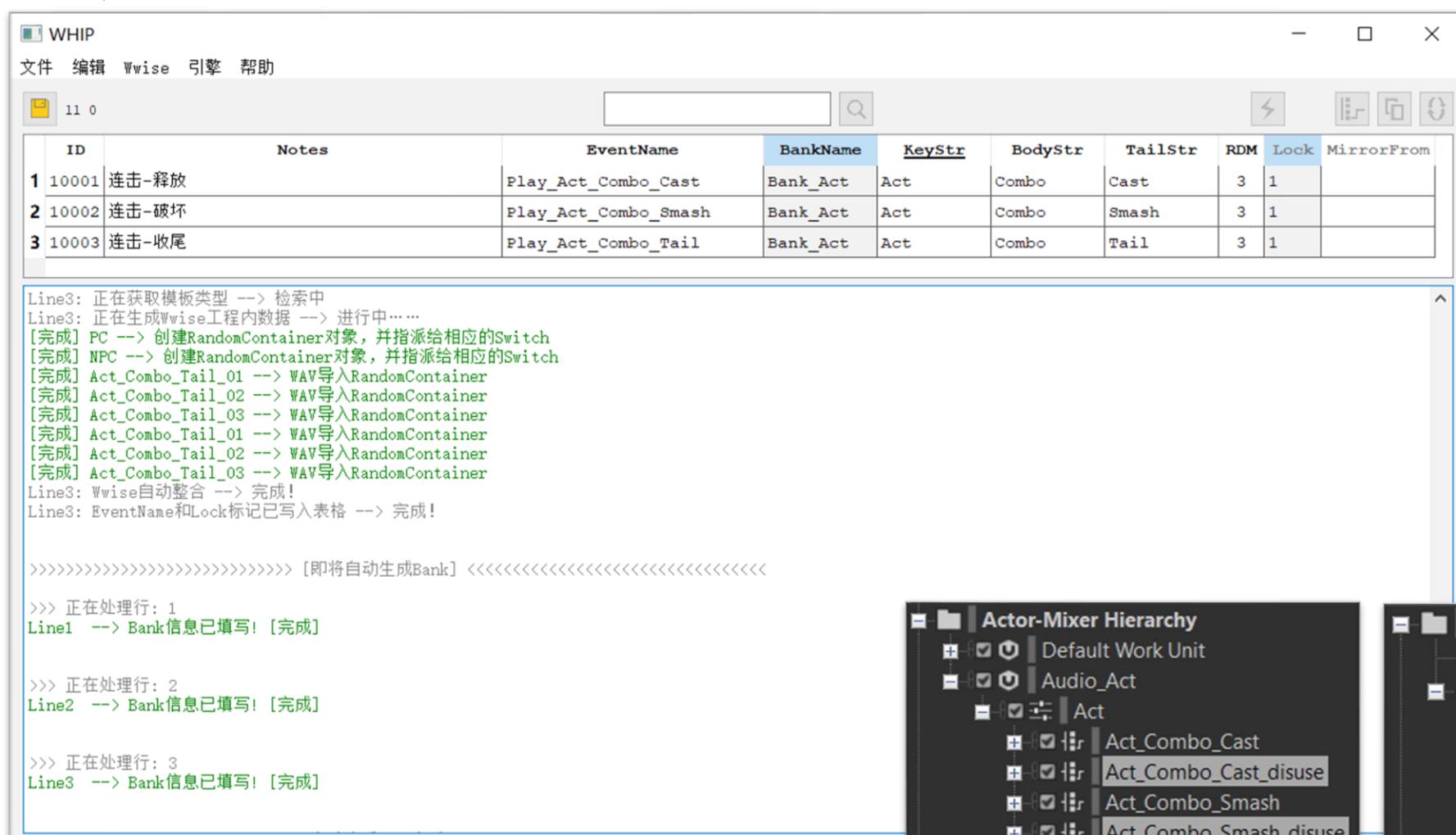
The screenshot shows the software interface after the regeneration process has been completed.

Table:

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1	10001 连击-释放	Play_Act_Combo_Cast	Bank_Act	Act	Combo	Cast	3	1	
2	10002 连击-破坏	Play_Act_Combo_Smash	Bank_Act	Act	Combo	Smash	3	1	
3	10003 连击-收尾	Play_Act_Combo_Tail	Bank_Act	Act	Combo	Tail	3	1	

The "BankName" and "Lock" columns for all rows have been updated, indicating the changes made by the regeneration process. The "EventName" column remains the same.

另起炉灶！AI上一轮生成的效果一般，再来一遍！（开发中，仅功能示意）

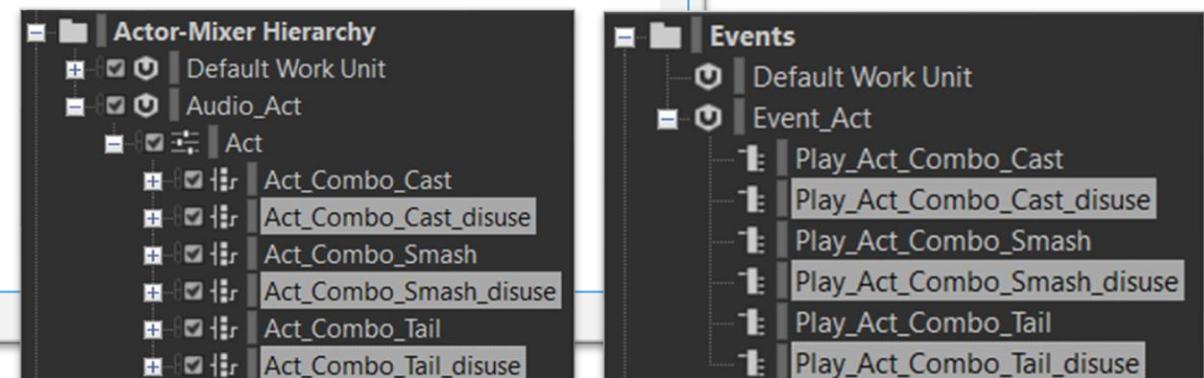


温馨提示：

由于当前的2.0版本未接入AI生成替条的功能，因此，该功能暂时仅能起到“单纯重新整合一遍”的效果。

之前的旧内容不删除，在原名称后添加“_disuse”，保留在工程中，以供与新内容参照比对。

另：该功能如果应对材质增加的情境，效果和“扩展 Switch对象”是一样的。

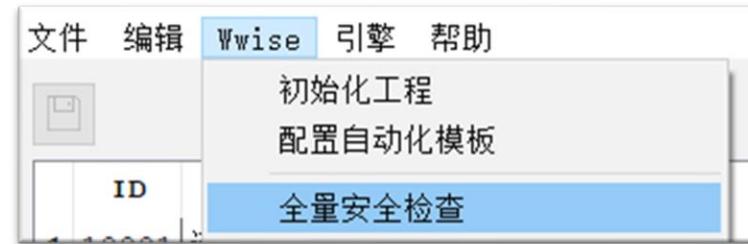


全量安全检查！

是否存在“尚未指派到Event”的容器对象?
是否存在“未写入表格”的Event事件?
是否存在“不属于当前Wwise工程”的自动化路径?
是否存在“未遵守结构规范”的对象?
.....

操作步骤：

点击设置菜单按钮，点击菜单栏中“Wwise → 全量安全检查”，可在日志台信息中获得详细的检查报告！例如下图：



ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001	连击-释放	Play_Act_Combo_Cast	Bank_Act	Act	Combo	Cast	3	1	
2 10002	连击-破坏	Play_Act_Combo_Smash	Bank_Act	Act	Combo	Smash	3	1	
3 10003	连击-收尾	Play_Act_Combo_Tail	Bank_Act	Act	Combo	Tail	3	1	

>>>>>>>>>>>>> 全量安全检查 - 开始 <<<<<<<<<<<<

```
[失联的Audio对象]NewTarget\Act_Combo_Cast_01.wav
[闲置的WAV对象]NewTarget\FT_Jump_Metal_03.wav
[闲置的WAV对象]NewTarget\Special_Amazing_Ele_gaga_03.wav
[闲置的WAV对象]NewTarget\Special_Amazing_Ele_swiii_Aha_02.wav
[闲置的WAV对象]NewTarget\FT_Gen_Water_Puddle_04.wav
[未填入表格的Event对象]{'Play_Act_Combo_Tail_disuse': '[FEA61030-A500-4EBD-ADC1-A3BB93E18A63']'}
[未填入表格的Event对象]{'Play_Act_Combo_Smash_disuse': '[3F761C1A-8D07-4071-B420-2316DD964A8F']'}
[未填入表格的Event对象]{'Play_Act_Combo_Cast_disuse': '[B2E02834-E577-4AA4-8DF0-AFB2077DA3D0']'}
```

>>>>>>>>>>>>> 全量安全检查 - 结束 <<<<<<<<<<<

全量安全检查!



【补充说明】

*插件在生成WWISE数据时，遵循着一系列严格的层级规范：

- 每一个KeyStr关键字的Audio WorkUnit，一定会产生在\Actor-Mixer Hierarchy根路径下。（禁止移动，禁止变更层级）
- 每一个类的父级Actor-Mixer，一定会产生在所属WorkUnit下（允许同层级移动，禁止变更层级）
- 每一个指派到Event的ObjectRef容器，一定会产生在所属\Actor-Mixer Hierarchy\WorkUnit(Audio_XXX)\Actor-Mixer(Audio_XXX)路径下（允许同层级移动，禁止变更层级）
- 每一个KeyStr关键字的Event WorkUnit，一定会产生在\Events根路径下（禁止移动，禁止变更层级）
- 每一个Event都会产生在所属的Event WorkUnit路径下（允许同层级移动，禁止变更层级）
- SoundBanks的对象生成路径规范也同理于Audio和Event（允许自由移动、允许自由变更层级）

*安全预警：切勿轻易更改插件自动化生成的对象的层级结构、关系，且强烈建议慎用“Virtual Folder”。

若对象的层级结构被打乱，安全检查功能将丧失判断的参考系，致使给出错误的、无意义的检查结果，还将破坏相关延伸功能模块的安全性、准确性！

为什么在设计插件时，要对层级结构限定如此严格的规范？

首先，这样的限制规范，完全不妨碍“音频内容的创造性发挥”。它的主要目的有两个：

- 1) 确保多人协作过程中的数据安全、可读性、易维护性。
- 2) 为延伸功能模块打基础，考虑到更进一步尝试扩展自动化能力边界的诉求。比如，未来“自动查bug”的条件基础。

关于多人协作。正因为WWISE允许任意一级的对象指派到Event，我们更需要警惕“无序”可能引发的更多不必要的时间浪费和麻烦！

比如，我们可以拿一个Switch容器造一个Play Event，但这个Switch容器下属的任意一个容器，也都可以直接造一个Play Event！

这自由度看似非常大对吧？但是呢，它是一把“双刃剑”。

WAV索引失联！怎么办？

有时候，我们想要挪动Originals/SFX路径下的wav文件，但一旦挪了位置，会导致Wwise内失联。

若想重新建联，可点击菜单栏中“Wwise → 失联WAV重关联”，确认弹窗提示后，即可完成Wwise内的全量扫描和重新连接，然后重新加载Wwise即可。



当然，如果是因为wav名称发生变更导致的失联，无法自动重新连接。将可能看到类似下图的提示。

温馨提示：

本插件2.0版本，选择将所有的wav都生成在了Originals/SFX路径下，不设置任何子文件夹，完全扁平。

原因是，在wav遵循着良好命名规范的前提下，对于多人协作、覆盖替换wav的便利性、插件使用流程、安全检查模块执行复杂度等几个方面综合考虑下来，这是相对最为平衡的选择。

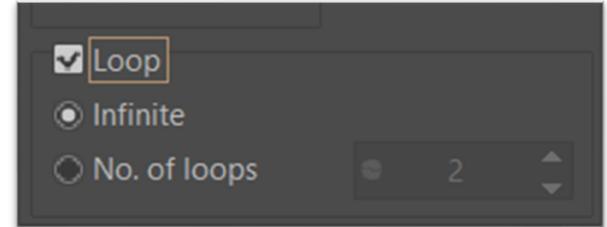
需要Loop的声音对象，一键组团，永不遗漏！



唯一步骤：

当命名的末尾是“_LP”这三个字符时，一键整合后，Wwise内，相关的Sound对象都将自动勾选Loop；插件数据表内将自动生成两个Event：一个以“Play_”开头，另一个以“Stop_”开头。

注：程序仅关注命名末尾是不是“_LP”，下图中ID 10001和ID 10002的两种写法都可以。



ID	Notes	EventName	BankName	<u>KeyStr</u>	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001				Act	Whoosh_LP		3		
2 10002				Act	Magic	LP	3		

整合完毕后，将自动新增两个Stop事件行，无需手动补充创建。

ID	Notes	EventName	BankName	<u>KeyStr</u>	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001		Play_Act_Woosh_LP	Bank_Act	Act	Whoosh_LP		3	1	
2 10002		Play_Act_Magic_LP	Bank_Act	Act	Magic	LP	3	1	
3 10003		Stop_Act_Woosh_LP						1	
4 10004		Stop_Act_Magic_LP						1	

需要Loop的声音对象，一键组团，永不遗漏！



【探讨】

为什么不自动为所有的Play Event都生成一个Stop Event?

答：没必要，也不需要。

想要停止某个Event的播放，除了通过Stop Event的方法之外，

程序端其实完全能够通过获取当前Event对象的playingID，使用stopPlayingID接口直接停止对应Event的播放，停止的方式和效果也可以控制。

之所以只给Loop类型的音效自动安排Stop Event，是因为通常情况下，Loop类音效表现需要被中止的时机和方式，直接交给设计师来安排，会更方便些罢了，当然也还是基于Game Object的前提下。但若说穿了，实际上也没必要。

*问题探讨：

其实，当某些无法依附于游戏内某个具体的Game Object，比如在地图坐标位置直接播放的Event，就难以通过Stop Event来停止了。

如果脱离Game Object，直接在某坐标触发同一个Play Event许多次（每一次的playingID都不同），Stop Event未必能轻易分辨究竟要停哪一个具体的Play Event对象。（除非Stop Global，把全场同一个Play Event都一刀切了）。

*安全预警（题外话）：

对于数据量庞大、多人协作的大型项目来说，在一个Event里同时包含多个不同逻辑指令的操作，可能也非常危险。

比如，一个Event里既有Play某个对象、也有Stop某个对象、或还有Set Switch之类的逻辑等，混杂交织在一起的形态……

尽管Wwise为设计师贴心地提供了这些“自由”“主动权”，但对于多人协作维护的项目来说，这却是一把“双刃剑”。

一些更适合由上游引擎端、客户端在源头上管理的逻辑，基于安全性、稳定性的考量，建议依旧交由上游来打理比较妥当。Wwise端Event的内容，需尽量保持“简单”。

如何一键导出xlsx需求表?



唯一步骤：

在插件主界面的表格数据中，选中需要导出的行。在选中的单元格区域上，点击鼠标右键，在弹出的菜单选项中，点击“导出需求表”。导出的xlsx需求表在工程所在文件夹的路径下。双击打开需求表，根据需要选择中文或英文内容，进一步编辑其中的内容细节即可。

*需求表的“样本命名”列中自动罗列出来的名称信息，对应的是每一条WAV的名字。

之所以要以每一条随机样本为最小单位全部罗列出来，一方面是为了方便做预算统计，另一方面是为了便于做进一步的单元格分类编辑。

The screenshot shows the software's main interface with a table of data. A context menu is open over rows 1, 2, 5, and 6, with the 'Export Requirements Table' option highlighted. Below the table is a detailed explanatory text about sample naming conventions. At the bottom, there is a table for managing samples, with columns for Sample Name, Channel Count, Reference Video Name, Content Description, Listening Requirement Description, Trigger Type (OneShot/Loop), Due Date, Feedback, and Other. The 'CN' button in the bottom toolbar is highlighted with a red box.

1 基本格式: 24bit 48kHz WAV; TruePeakLimit: -3dBFS; 平均响度: -16 (+-2)LUFS
2 起始帧位置: 可参考用户点击触发行为反馈的瞬间, 或可参考视频自带音轨中的pop位置 (若有的话)。若两者时间位置不同, 则以pop位置为准。
3 随机样本: 随机样本的数量已直接体现在第一列“样本命名”当中, 命名尾部带有“_01”、“_02”、“_03”字样的, 表示属于同一个动作的多个随机样本。尾部不带有“_xx”的, 例如, 1234_skill01, 表示这个动作仅涉及制作一条样本。
4 涉及分轨的样本命名结构: “样本命名_分轨关键字” (例: 1234_show_amb; 1234_show_spfx; 1234_show_foley……)
5 回包压缩包命名: “1234_SFX.zip” (压缩包中需包含所有的wav文件, 和贴上音轨的视频文件。若视频中同一个动作涉及多次触发, 任选一段贴合即可, 无需贴满整个视频)
6 基本描述: 角色是XXXX, 它的特征是XXXXX……
7 听觉表现参考: (贴链接)
8 样本数总计: XX

ID	Notes	EventName
1 10001	技能-Whoosh-循环	播放Event
2 10002	技能-魔法-循环	停止Event
3 10003		展开Event
4 10004		更新Event的Notes
5 10005	技能-万籁俱寂-a段	填充Bank信息
6 10006	技能-万籁俱寂-b段	打开资源文件夹
		检查替条
		设置文字颜色
		设置单元格底色
		导出需求表

	样本命名 (以随机样本为最小单位, 方便一键求和统计样本数量)	声道数 Mono/Stereo	参考视频名	内容描述	听感需求描述或关键词	单发/循环 (Oneshot/Loop)	最晚提交日期	需求反馈	其他
11	Act_Woosh_LP_01			技能-Whoosh-循环					
12	Act_Woosh_LP_02								
13	Act_Woosh_LP_03								
14	Act_Magic_LP_01			技能-魔法-循环					
15	Act_Magic_LP_02								
16	Act_Magic_LP_03								
17	Act_silence_a_01			技能-万籁俱寂-a段					
18	Act_silence_a_02								
19	Act_silence_b_01			技能-万籁俱寂-b段					

其他辅助功能和使用技巧——音频简表导出 (窗口顶部菜单栏, 点选“文件→显示/隐藏偏好设置”)



【导出客户端读取用的“音频简表”】

主界面中的所有信息，都保存在一个json文本中，不过对于客户端侧读取而言，仅需要ID、EventName、BankName这三个信息就够了。因此，我们可以先在偏好设置的“音频简表导出路径（客户端侧读取用）”中指定一个引擎路径，然后通过菜单栏“文件→导出音频表”导出即可。

The screenshot shows the WHIP application window with the "File" menu open. The "Preferences" option is selected, highlighted with a red arrow. In the preferences dialog, there is a section for "音頻簡表導出路徑 (客戶端側讀取用)" (Export Sound List Path (Client-side readable)) where the path "C:\Users\admin\Desktop" is entered. This path is also highlighted with a red box.

EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
Play_Amb_CC	Amb	CC			3		
Play_FT_DD	FT	DD			4		
Play_VO_EE	VO	EE			5		
Play_Act_FF	Act	FF			6		
Play_UI_GG	UI	GG			7		
Play_Amb_HH	Amb	HH			8		

```
{
    "$ProjectStr$": "Sample.wproj",
    "Data_SoundList": [
        "10001": {
            "Notes": {
                "text": "雷欧奥特曼"
            },
            "EventName": {
                "text": "Play_Act_AA"
            },
            "BankName": {
                "text": ""
            }
        },
        "10002": {
            "Notes": {
                "text": "皮卡丘"
            },
            "EventName": {
                "text": "Play_UI_BB"
            },
            "BankName": {
                "text": ""
            }
        },
        "10003": {
            "Notes": {
                "text": "擎天柱"
            }
        }
    ]
}
```

其他辅助功能和使用技巧——自动打Bank (窗口顶部菜单栏, 点选“文件→显示/隐藏偏好设置”)



【自动打Bank】

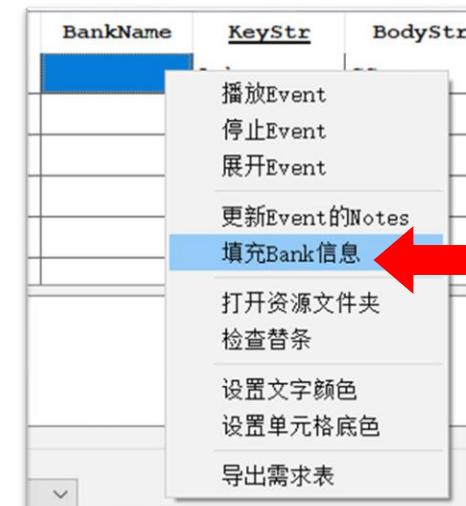
如果给“数据整合完毕后，顺便自动打SoundBank”选择“是”，“一键整合”“扩展Switch对象”“镜像ID”“另起炉灶”四个功能模块在完成后会自动打Bank，并将结果依次自动填充到表格里BankName单元格中。如果选择“从不”，就不会自动打。

The screenshot shows the WHIP software interface. The top menu bar includes File, Edit, Wwise, Engine, and Help. The 'File' menu is currently selected. A sub-menu under 'File' contains options like Load Audio Table, Save, and Preferences. The 'Preferences' tab is active, showing settings for language (Simplified Chinese), startup behavior (never), and data integration preferences. A red box highlights the dropdown for 'Automatically create SoundBank after data integration' which is set to 'Never'. Below this, there's a table view with columns: EventName, BankName, KeyStr, BodyStr, TailStr, RDM, Lock, and MirrorFrom. The table contains several rows of event data.

温馨提示:

如果Bank是后来手动打的，可以选中所有没有填写BankName信息的行，右键点选“填充Bank信息”。

程序将自动扫描每一个Event所在的位置，如果定位到Bank结果，会自动填写到单元格中。



其他辅助功能和使用技巧——Event上色



【全量安全检查后，给在Wwise中但不在单元格中的Event上色】

如果选“是”，通过点选菜单栏“Wwise → 全量安全检查”后，若发现Wwise工程中没有填写到表格中的Event，会自动将Event的颜色标记变为粉红色。

The screenshot shows the Wwise WHIP interface. In the top menu bar, the 'Wwise' tab is selected. On the left, there's a sidebar with various options like '初始化工程' (Initialize Project), '配置自动化模板' (Configure Automation Template), and '全量安全检查' (Full Safety Check). The main area displays a table with columns: EventName, BankName, KeyStr, BodyStr, TailStr, RDM, Lock, and MirrorFrom. Below the table, there's a large empty text input field. At the bottom, the '偏好设置' (Preference Settings) dialog is open, containing several configuration options. A red box highlights the '全量安全检查后，给在Wwise中但不在单元格中的Event上色' (Color Events in Wwise that are not in the table after full safety check) dropdown, which is set to '是' (Yes). A yellow arrow points from this setting to the 'Event' table, where a row for 'Play_VO_OO' is highlighted in pink, indicating it was not found in the table. To the right, a separate window shows the 'Event' table with a single row for 'Play_VO_OO'. The status bar at the bottom indicates a full safety check is starting.

温馨提示：

这个功能，低于2019.2.8的Wwise版本不支持。

其他辅助功能和使用技巧——常规偏好设置 (窗口顶部菜单栏, 点选“文件→显示/隐藏偏好设置”)



其他功能:



语言/Language: 用来切换主界面的显示语言。目前只提供了中文和 English 两个选项。

启动时显示欢迎界面: 选择“是”，则每次启动时都会先弹出欢迎界面。选择“从不”，则每次启动时直接进入主界面。

启动时自动加载音频表: 选择“是”，则音频表json文件将在启动时自动加载，在主窗口中显示表格；选择“从不”，则主窗口启动后，不会显示加载JSON并显示表格，需要手动通过主界面“文件→加载音频表”来加载。

复制/粘贴/清除 模式: 根据用户选择的内容，来处理单元格内的文本和颜色信息

每隔几分钟，自动另存为.json一次: 这里如果选择“是”，并设置一个时间，则工具会根据设定的时间，定时自动保存完整的JSON作为快照备份

日志显示内容: 其中有几个不同的档位。如果选择显示的档位越少，则能够在日志台中看到的执行信息类型就越少。建议保持“提示/警告/报错/高危”这一档。

其他辅助功能和使用技巧——快速验证Event状态



【使用“播放Event、停止Event、展开Event”快速验证Event状态】

选中任意行的EventName，右键点选“播放Event”或“停止Event”，则可直接播放和停止Event的内容，无需回到Wwise内翻找。此时Wwise内也会直接将当前在正在播放的Event显示在前台。

若点选“展开Event”，插件主界面下方将会出现一个新的区域。区域中会显示该Event的Action、以及每个Action中ObjectRef对应的wav的波形。我们可以在波形上的任意位置双击鼠标左键，则将直接从双击的位置开始播放原始wav的内容。如果wav较长，需要中途停止播放，则右键点击波形任意位置即可。

注：有一个插件2.0版本未呈现的功能：展开Event后，除了能显示波形信息之外，还可以显示当前Event的“完播率分析”。

比如，将影响Event播放的因素，按照信号流从前到后的顺序，呈现出一个分析简报，并提示可能出bug的关节。

The screenshot shows the Event Manager interface with a context menu open over the row for Event 11 (Play_Act_KK). The menu options are:

- 播放Event (Play Event)
- 停止Event (Stop Event)
- 展开Event (Expand Event) - This option is highlighted.
- 更新Event的Notes (Update Event Notes)
- 填充Bank信息 (Fill Bank Information)
- 打开资源文件夹 (Open Resource Folder)
- 检查替条 (Check Substitutes)
- 设置文字颜色 (Set Text Color)
- 设置单元格底色 (Set Cell Background Color)
- 导出需求表 (Export Requirements Table)

At the bottom of the interface, there are status bars and a waveform preview. A green arrow points to the waveform preview area.

Event ObjectRef: 1 Play --> SwitchContainer --> Act_KK

WAV: 3 C:\Users\admin\Desktop\Sample\Originals\SFX\Act_KK_01.wav 定位当前的WAV

其他辅助功能和使用技巧——Event如何一键更新Notes?



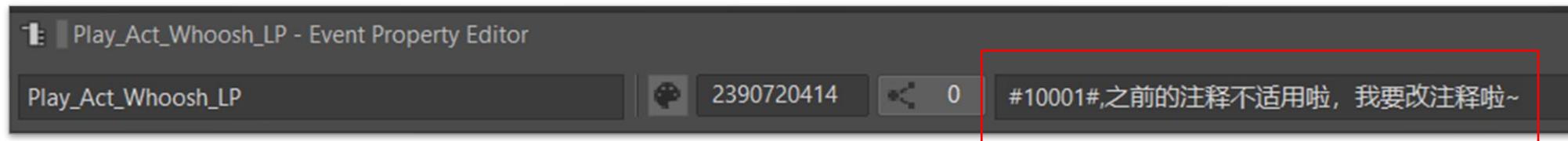
唯一步骤：

在插件主窗口数据表中，在Notes列所在的单元格中，输入新的Notes。

在更改后的Notes单元格上，点击鼠标右键，在弹出的菜单选项中，点击“更新Event的Notes”。可多选一次性批量处理多行。

Event名无误的情况下，Wwise工程中相应的Event的Notes会立即完成更新。

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1 10001	之前的注释不适用啦，我要改注释啦~	Play_Act_Whoosh_LP	Bank_Act	Act	Whoosh_LP		3	1	
2 10002		Play_Act_Magic_LP	Bank_Act	Act	Magic	LP	3	1	
3 10003		Stop_Act_Whoosh_LP						1	
4 10004		Stop_Act_Magic_LP						1	



其他辅助功能和使用技巧——如何检查替条是否都更换过正式资源了、是否有遗漏？

唯一步骤：

选中需要确认的行，选中任意一个或多个单元格都行，点击鼠标右键，在菜单选项中，点击“检查替条”。

如果当前行MirrorFrom列的单元格背景色变成淡红色了，说明这个Event相关的WAV尚且存在未经覆盖、替换过的静音替条（日志台中也会显示具体信息）；反之，如果MirrorFrom列的单元格没有变色，说明这个Event下属所有的WAV都已不再是静音替条。

The screenshot shows a software interface for managing audio events. A table lists various events with columns for ID, Notes, EventName, BankName, KeyStr, BodyStr, TailStr, RDM, Lock, and MirrorFrom. Rows 5 and 6 have their EventName cells selected, and a context menu is open over row 5. The menu includes options like '播放Event' (Play Event), '停止Event' (Stop Event), '展开Event' (Expand Event), '更新Event的Notes' (Update Event's Notes), '填充Bank信息' (Fill Bank Information), '打开资源文件夹' (Open Resource Folder), '检查替条' (Check Placeholder) which is highlighted with a blue background, '设置文字颜色' (Set Text Color), '设置单元格底色' (Set Cell Background Color), and '导出需求表' (Export Requirements Table). Red arrows point from the text '检查替条' to both the menu item and the corresponding column in the table header. The MirrorFrom column is the last one on the right.

ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
1	10001	Play_Act_Whoosh_LP	Bank_Act	Act	Whoosh_LP		3	1	
2	10002	Play_Act_Magic_LP	Bank_Act	Act	Magic	LP	3	1	
3	10003	Stop_Act_Whoosh_LP						1	
4	10004	Stop_Act_Magic_LP						1	
5	10005	Play_Act_slience_a	Bank_Act	Act	slience	a	2	1	
6	10006	Play_Act_slience_b			slience	b	3	1	

--> 5 Play_Act_slience_a
[提示] 这一行的Event所相关的WAV里包含静音替条 --> 5
C:\Users\admin\Desktop\Sample\Originals\SFX\Act_slience_a_01.wav
C:\Users\admin\Desktop\Sample\Originals\SFX\Act_slience_a_02.wav

--> 6 Play_Act_slience_b
[提示] 这一行的Event所相关的WAV里包含静音替条 --> 6
C:\Users\admin\Desktop\Sample\Originals\SFX\Act_slience_b_01.wav
C:\Users\admin\Desktop\Sample\Originals\SFX\Act_slience_b_02.wav
C:\Users\admin\Desktop\Sample\Originals\SFX\Act_slience_b_03.wav

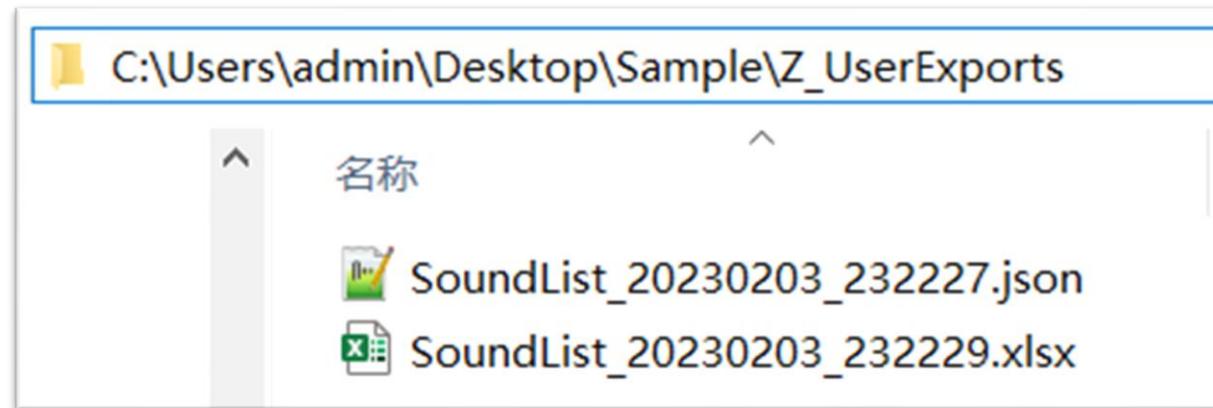
其他辅助功能和使用技巧——另存为JSON、xlsx



【另存为JSON、xlsx】

通过点选菜单栏“文件→另存为.json”，可以将当前表格中的所有数据，导出一个.json文件副本，保存在当前Wwise工程路径“Z_UserExports”文件夹下。通过点选菜单栏“文件→另存为.xlsx”，可以将当前表格中的所有数据，导出一个.xlsx文件副本，保存在当前Wwise工程路径“Z_UserExports”文件夹下。

注：另存为的文件名，是“SoundList”和当前时间“年月日时分秒”信息的拼接。因此，我们可以从文件名判断文件生成的初始时间。



注：Wwise工程路径下的“Z_UserExports”文件夹，是由工具自动产生，用来存放用户另存为的JSON、xlsx等临时数据文件的。

其他辅助功能和使用技巧——xlsx → JSON



【xlsx --> json】

我们可以将一张符合插件格式要求的xlsx音频表，转换为一张可以被插件直接加载为表格的JSON文本，内容包括单元格中的文本、文本颜色、单元格背景色。

首先，我们在菜单栏中点选“文件 → xlsx --> json”，选择需要转表的xlsx。

如果xlsx是一个有效的xlsx，则转换后的JSON文件，将保存在需要将当前Wwise工程路径“Z_UserExports”文件夹下。

JSON文件的命名，是“SoundList”和当前时间“年月日时分秒”信息的拼接。

xlsx表的格式要求：

- 1) 首行单元格内的文本和顺序，必须和插件中表格的表头完全一致
- 2) 正式的数据内容，从第2行开始逐行记录

	A	B	C	D	E	F	G	H	I	J
1	ID	Notes	EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFrom
2	10001	雷欧奥特曼	Play_Act_AA		Act	AA		1	1	
3	10002	皮卡丘	Play_UI_BB		UI	BB		2	1	
4	10003	擎天柱	Play_Amb_CC		Amb	CC		3	1	
5	10004	水冰月	Play_FT_DD		FT	DD		4	1	
6	10005	亚古兽	Play_VO_EE		VO	EE		5	1	
7	10006	孙悟饭	Play_Act_FF		Act	FF		6	1	
8	10007	那路透	Play_UI_GG		UI	GG		7	1	
9	10008	江户川柯南	Play_Amb_HH		Amb	HH		8	1	
10	10009	那路透萨斯给	Play_FT_II		FT	II		1	1	
11	10010	怪盗基德	Play_VO_JJ		VO	JJ		2	1	
12	10011	四驱兄弟	Play_Act_KK		Act	KK		3	1	
13	10012	灌篮高手	Play_UI_LL		UI	LL		4	1	
14	10013	鲁邦三世	Play_Amb_MM		Amb	MM		5	1	
15	10014	光能使者	Play_FT_NN		FT	NN		6	1	
16	10015	圣斗士			VO	OO		7	1	



```
$ProjectStr$": "Sample.wproj",
"Data_SoundList": {
  "10001": {
    "ID_textColor": "#000000",
    "ID_bgColor": "#ffffffff",
    "Notes": {
      "text": "雷欧奥特曼",
      "textColor": "#000000",
      "bgColor": "#ffffffff"
    },
    "EventName": {
      "text": "Play_Act_AA",
      "textColor": "#000000",
      "bgColor": "#ffffffff"
    },
    "BankName": {
      "text": "",
      "textColor": "#000000",
      "bgColor": "#ffffffff"
    },
    "KeyStr": {
      "text": "Act",
      "textColor": "#000000",
      "bgColor": "#ffffffff"
    },
    "BodyStr": {
      "text": "AA",
      "textColor": "#000000",
      "bgColor": "#000000"
    }
  }
}
```

其他辅助功能和使用技巧——基于工程内容生成JSON音频表



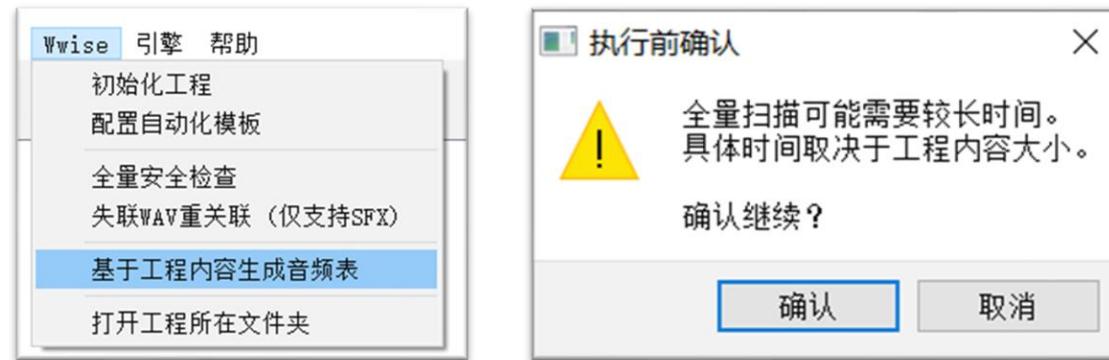
【基于工程内容生成JSON音频表】

如果我们第一次使用插件，插件本身初始化生成的info.json文件中，并没有记录任何有用的数据。

如果此时，Wwise里已经有了大量的数据，这时候，如果我们想要插件接管Wwise，则需要先将Wwise内部的Event全部扫描一遍，生成一张可用的JSON文本。

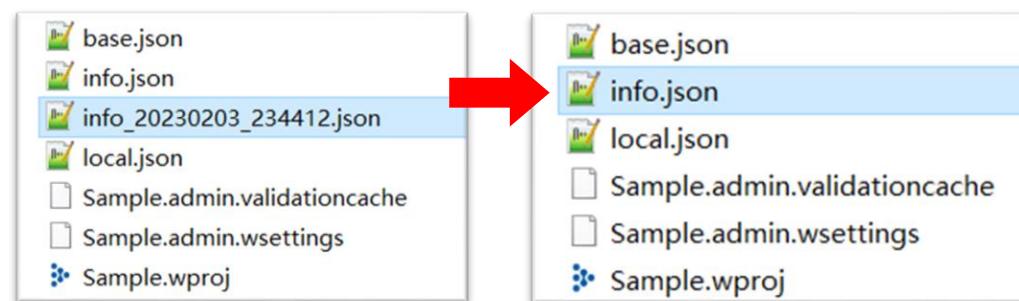
第一步：点选菜单栏中“Wwise→基于工程内容生成音频表”。此时将弹出一个确认窗口，没问题的话点击确认。然后静待执行即可。

注：如果Wwise工程中的Event数量很多，这个过程可能会导致插件窗口“假死”。这时候什么都不要动，等待到执行完毕、窗口恢复正常即可。



第二步：新生成的json文本，将直接出现在Wwise工程的根目录下，命名格式为“info”和“时间信息”的拼接。

此时，**手动将旧的空白的info.json文件删除**，将这个新生成的json文件的命名，直接**手动更改为“info.json”**，然后**重启插件**即可。



其他辅助功能和使用技巧——检查音频ID挂接状态

【检查音频ID挂接状态】

如果引擎端可以收集并统计全局所有音频ID的配置位置，并与音频简表做一个对比，将对比后的差异按照下图中的格式，保存到一个JSON文本里。

```
{  
    "ID_ExistInSoundList_NotExistInEngine": ["ID1"],  
    "ID_ExistInEngine_NotExistInSoundList": ["ID2"],  
    "ID_FoundDuplicate_InEngine": {  
        "ID3": [  
            "location1",  
            "location2"  
        ]  
    },  
    "Event_ExistInSoundList_NotExistInEngine": ["Play_01"],  
    "Event_ExistInEngine_NotExistInSoundList": ["Play_02"],  
    "Event_FoundDuplicate_InEngine": {  
        "Play_03": [  
            "location1",  
            "location2"  
        ]  
    }  
}
```

>>>>>>>>>>>>>>>> 音频ID挂接状态报告 <<<<<<<<<<<<<<<

[警告] 发现音频表中有、但尚未挂接到版本中的ID！

→ ID1
[警告] 出现版本与本地校验不一但本地未更新的文件

[警告] 发

[警告] 发现出现在了多个挂接位置的同一个TD！

```
[{"ID3": [ "location1", "location2"]}]
```

[警告] 发现音频表中有、但尚未挂接到版本中的Event！

--> Play C

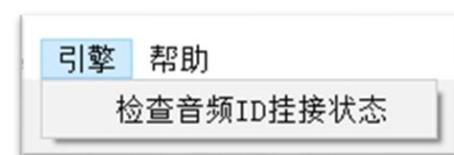
[警告] 发现版本中有挂接记录、但音频表里却不存在的Event！

--> Play_

[警告] 发现出现在了多个挂接位置的同一个Event!

```
        "Play_03": [
            "location1",
            "location2"
        ]
    }
```

我们便可以通过在菜单栏中点选“引擎→检查音频ID挂接状态”，将对比报告显示在日志台中，查看音频ID的实际挂接情况。



其他辅助功能和使用技巧——过滤器



【使用“过滤器”分类显示数据】

当表格内数据量较多时，我们可以通过菜单栏，点击“文件→隐藏/显示 过滤器”，过滤器将出现在窗口顶部。
设计师可通过点选一个或几个类，仅显示选择的相关内容、隐藏非相关内容。

The screenshot shows the WHIP software interface. On the left, a vertical menu bar lists various file operations like saving, exporting, and filtering. The '显示/隐藏 过滤器' (Show/Hide Filter) option is highlighted with a blue selection bar. In the center, there's a main data table with columns: EventName, BankName, KeyStr, BodyStr, TailStr, RDM, Lock, and MirrorFro. Above the table, there's a toolbar with several icons. A red box highlights the dropdown filter button in the toolbar. To the right of the main table, a smaller window displays the 'BankName' column's filter dialog. This dialog shows a dropdown menu with options: All, UI, Act, Amb, VO, and FT. The 'UI' option is selected and highlighted with a blue selection bar. The main table's 'BankName' column now only displays rows where the value is 'UI'.

EventName	BankName	KeyStr	BodyStr	TailStr	RDM	Lock	MirrorFro
10 10010		Act					
11 10011		UI					
12 10012		Amb					
13 10013		FT					
		VO					
		Act					
		UI					
		Amb					
		FT					
		VO					
		Act					
		UI					
		Amb					

其他辅助功能和使用技巧——搜索文本



【使用“搜索”查找文本】

在主界面正上方的文本框中输入需要搜索的文本，点击搜索按钮，包含目标文本的单元格，将会呈现高亮的状态，显示在表格中。如果有多个结果，连续点击搜索按钮，会自动在多个结果中轮流切换显示。

The screenshot shows a table with 10 rows of data. The columns are labeled: ID, Notes, EventName, BankName, and Key. A search bar at the top contains the text "那路透". A red arrow points from the search bar to the magnifying glass icon. A green double-headed arrow highlights the row for ID 10007, which contains the text "那路透". Another green double-headed arrow highlights the row for ID 10009, which also contains the text "那路透".

ID	Notes	EventName	BankName	Key
1	10001 雷欧奥特曼	Play_Act_AA	Bank_Act	Act
2	10002 皮卡丘	Play_UI_BB	Bank_UI	UI
3	10003 摧天柱	Play_Amb_CC	Bank_Amb	Amb
4	10004 水冰月	Play_FT_DD	Bank_FT	FT
5	10005 亚古兽	Play_VO_EE	Bank_VO	VO
6	10006 孙悟饭	Play_Act_FF	Bank_Act	Act
7	10007 那路透	Play_UI_GG	Bank_UI	UI
8	10008 江户川柯南	Play_Amb_HH	Bank_Amb	Amb
9	10009 那路透萨斯给	Play_FT_II	Bank_FT	FT
10	10010 怪盗基德	Play_VO_JJ	Bank_VO	VO

其他辅助功能和使用技巧——设置文字颜色、单元格底色



【设置文字颜色、单元格底色】

我们可以像在Excel中一样，编辑文本的颜色和单元格底色。

仅需选中需要编辑的单元格，右键点选“设置文字颜色”或“设置单元格底色”，选择颜色，确认后生效。

注：颜色信息，将跟随文本信息，一起保存在info.json中。

ID	Notes	EventName
3	擎天柱	Play_Amb_CC
4	水冰月	
5	亚古兽	
6	孙悟饭	
7	那路透	
8	江户川柯南	
9	那路透萨斯给	
10	怪盗基德	
11	四驱兄弟	
12	灌篮高手	
13	鲁邦三世	
14	光能使者	

擎天柱
水冰月
亚古兽
孙悟饭
那路透
江户川柯南
那路透萨斯给

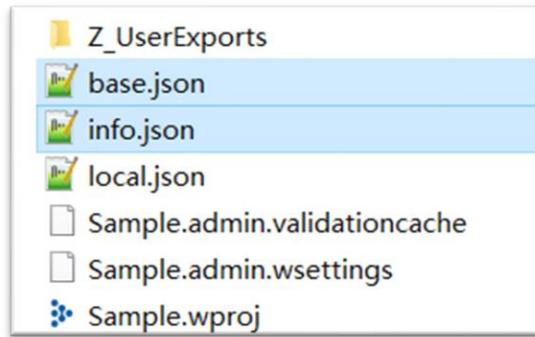
其他辅助功能和使用技巧——版本管理



插件在运行中会产生一些文件。

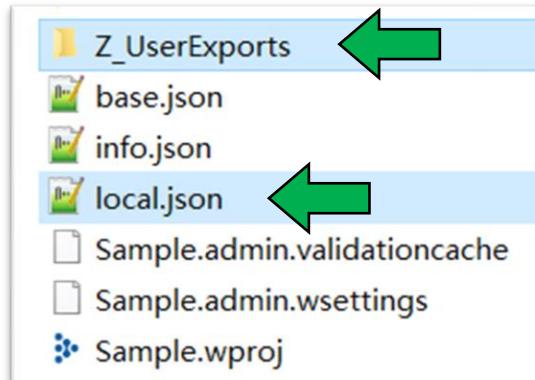
需要参与版本管理 ([Perforce](#)、[UGIT](#)、[SVN](#)等)，跟随Wwise工程数据一同上传、下载，参与多人协作同步更新、比对差异、处理冲突的有：

- 1) Wwise工程根目录下的2个json文件：“[info.json](#)” “[base.json](#)”
- 2) 导出到引擎路径下，提供给客户端读取用的音频简表 “[SoundList.json](#)”



不需要参与版本管理，建议添加到[忽略列表](#)里的有：

- 1) Wwise工程根目录下的“[local.json](#)”，和1个插件产生的文件夹：“[Z_UserExports](#)”
- 2) 插件自身目录下的所有文件

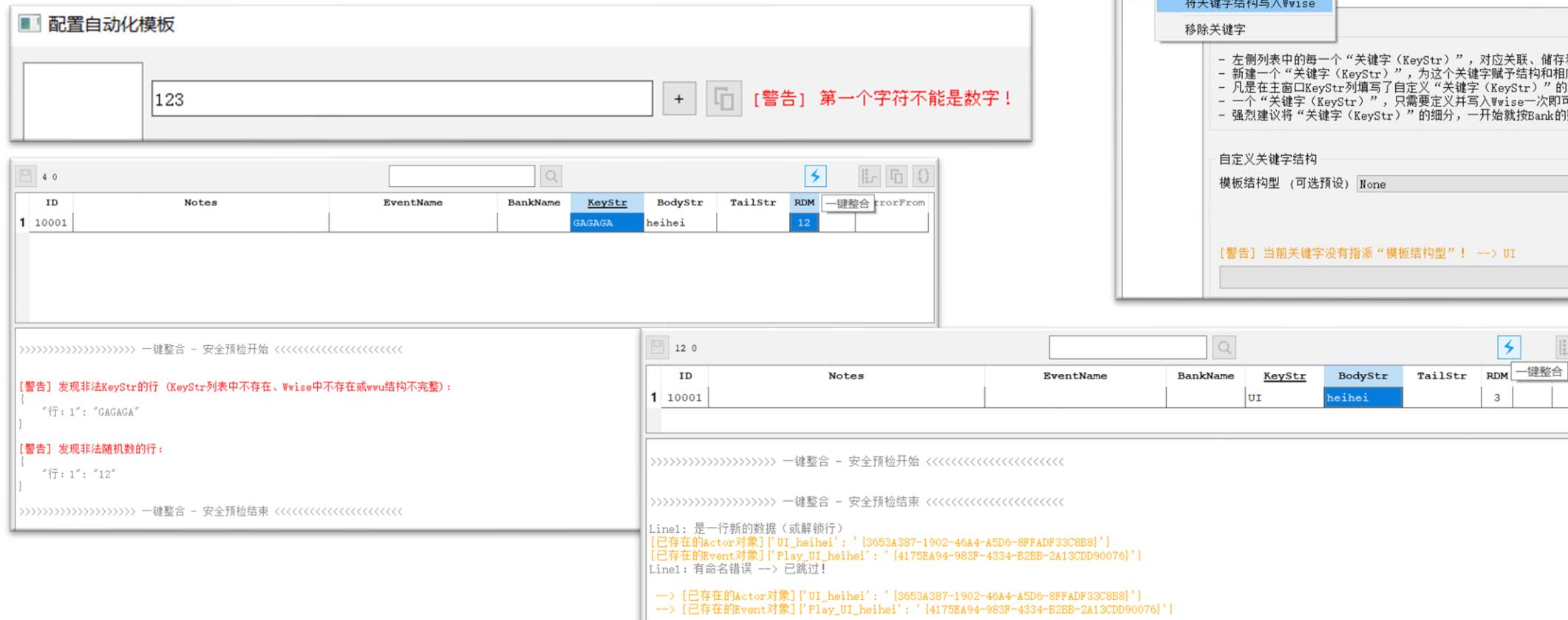


其他辅助功能和使用技巧——报错说明

插件在正常运行时，我们可以在日志台中看到许多常规、友好的日志和报告。
但插件运行异常、或检测到用户可能有操作失误的时候，也可能会出现一些警告类的**安全提示**。

为辅助设计师更顺畅地使用插件，开发者在各个角落里尽可能安排了足够多的安全提示。
高亮显示的字体颜色、警告窗口等信息，基本上都能直接吸引到用户的注意力，以便让用户能快速察觉到可能存在的操作安全问题。

由于插件内设置的安全提示信息非常非常多，说明书暂不一一赘述了，仅举一些示例以供参考：





“隐形杂务缠身，无心专注创作”

近些年，身边越来越多的设计师朋友开始议论这种困扰。

不同公司的人力、财力各不相同，外延环境所带来的工作代偿，即便是已经在努力扭转局面的Wwise，恐怕也爱莫能助。况且，设计师的工作任务、压力等级、舒适度，很大程度上也取决于项目的实际运营和管理状况。

以“管事务”为重心的管理者，更关心设计师在完成项目需求时，是否能“随机应变”“按时完成”“解决隐患”。

以“管数据”为重心的管理者，更关心设计师产出的大量数据是否“完备规整”“清晰简洁”“易维护”“易交接”。

以“做业绩”为重心的管理者，更关心设计师是否能制造业务“亮点”，或者是否能第一时间解决掉上级的迫切需求。

设计师面对的管理者，可能是以上其中一种，也可能是几种的组合。

不同的企业文化、体制、组织架构，对管理者的要求不同，继而，管理者对设计师的态度，自然也“千式百样”。

相应地，不同设计师的工作重心、价值取向、是非、成败、荣辱，也因此大相径庭。

那么，有多少设计师可以做到左右逢源呢？

或者，有多少设计师可以在做到左右逢源的同时，还总能设计出“精彩的作品”呢？

现实里，并不是所有人都能做选择。

若设计师总是受困于隐形杂务干扰，难以专注于真正意义上的创作，不论对于个人、团队、还是行业，终究都不是长久之计。

Wwise将游戏音频业务方方面面的细枝末节，做足了归纳梳理，为设计师提供了开阔的操作视野。

那么“Wwise闪电分身”则继续接力，抛砖引玉，与设计师朋友们一同探讨：如何把平日里被偷走的时间，通通拿回来！

如果总是疲于奔命，那人生可要错过太多有趣的风景了，不是么？