ADDING SOUNDS TO SOUNDMINER – A SYSTEM BY TIM NIELSEN & JUSTIN DRURY

THE FILENAME CONVENTION

This system uses a very specific four or five part filename, that if adhered to, allows automated scripts in Soundminer, which utilize a lookup table _facilitylookup.lua and a category list _categorylist.csv, to automatically parse out information from the filename into various fields in Soundminer.

The filename requires at least four parts, but may include five, but each block must be separated by an _ and no other _s must be used. The _ is used to separate the blocks as follows:

CatID_FXName_ShortID_Source_UserData

CatID = Abbreviated Category / Subcategory (defined by the _categorylist.csv file)
FXName = Brief Description (under 25 characters preferably)
ShortID = Sound Designer's Initials (matched using _facilitylookups.lua file)
Source = Abbreviated Show Title (matched using _facilitylookups.lua file)
UserData = A User defined space, often used for ID or Number, usually used for guaranteeing that the Filename is 100% unique… or storing things such as microphone type, location, etc.

For example:

BGPub_Busy Interior Restaurant Bar Walla Evening 01_TN_DORY_1024.wav

Is a valid filename because they contain all five parts separated by _s

Adhering to this filename structure allows the use of 'scripts' within Soundminer to automatically fill in the most commonly used fields. There are instructions at the end of this document explaining how to use the scripts to help you in adding Metadata to your records.

The **CatID** represents both the Category and SubCategory as defined in the file _categorylist.csv This list is customizable. It is based by default on the Universal Sound Category List made by Tim Nielsen and available as an Excel spreadsheet as well. This list contains three main columns, **Category**, **SubCategory**, and **CatID**, which is an abbreviated tag that contains info for both the **Category** and **SubCategory**:

An example is shown below:

Category	SubCategory	CatID
AIR	BLOW	AIRBlow
AIR	BURST	AIRBrst
AIR	HISS	AIRHiss
AIR	MISC	AIR
AIR	STEAM	AIRStm
AIR	SUCTION	AIRSuck

The _categorylist.csv file also may contain other fields, such as the foreign translation lists, and also contains an 'explanations' column that has no effect in Soundminer, but is meant to help clarify some fields that might be confusing. Again, more on the using of this table and the scripts at the end of this document.

The next chunk, what we're calling **FXName** is what will become the **FXName** field in Soundminer. The goal is to give a brief description of the sound. Around 25 characters is usually ideal for the length of this field. It's important that for the scripts to work properly, that **FXName** be 'TitleCased' if you are not using spaces in this block. In other words, each new word receives a Capital at the beginning. So in the above example:

BGPub_BusyInteriorRestaurantBarWallaEvening01_TN_DORY_1024.wav

The FX Name would end up BUSY INTERIOR RESTAURANT BAR WALLA EVENING 01

But:

BGPub_Busyinteriorrestaurantbarwallaevening01_TN_DORY_1024.wav

Would generated the FX name: Busyinteriorrestaurantbarwallaevening01, which is not ideal.

If separating words with spaces, this rule doesn't apply.

ShortID shows you who recorded or designed the sound. These will be stored in the _facilitylookups.lua file, You will need to manually modify the _facilitylookup.lua file to add matches here. See the end of this document for modifications to the file.

If the script makes a match here, it will automatically fill in the DESIGNER field with the match.

Source holds the agreed upon initials or code for your show or project. This is also stored in _facilitylookups.lua, and you can enter in your own abbreviations and full show or project name. If the script finds a match here, it will parse out into the SHOW field.

A few quick notes about filenames:

- There should be three or four _ in the filename, as they separate the various parts of the name for the scripts to work properly. Please do not use additional underscores. And the first four chunks of data must be **CatID_FXName_ShortID_Source**… the last _ and **UserData** is optional. The scripts will ignore this last chunk of data. You could easily build a workflow to take this piece of information and place in a designated field. We may look at modifying the scripts to do this later too.

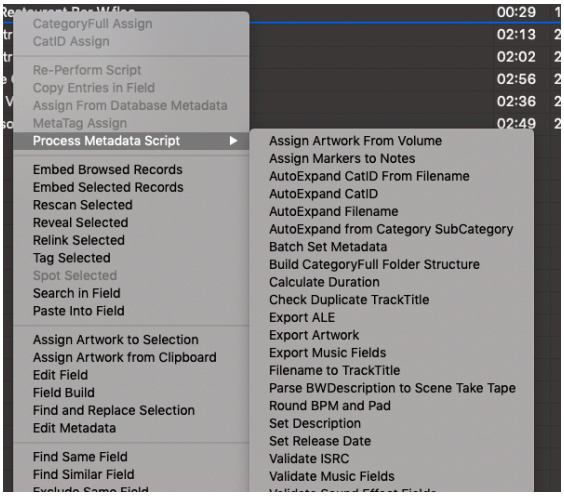
- If you need to number sound effects, please do it as the last part of the **FXName** part of the filename. For example, if you have four versions of the example above, they should be named:

BGPub_BusyInteriorRestaurantBarWallaEvening01_TN_DORY_1024.wav
BGPub_BusyInteriorRestaurantBarWallaEvening02_TN_DORY_1025.wav
BGPub_BusyInteriorRestaurantBarWallaEvening03_TN_DORY_1026.wav
BGPub_BusyInteriorRestaurantBarWallaEvening04_TN_DORY_1027.wav

USING THE SCRIPTS TO PARSE INFORMATION FROM THE FILENAME

Assuming you have named your filename strictly according to the instructions above, Soundminer has a powerful feature in the form of .lua scripts, to assist you in adding Metadata to your records.

Scripts are accessed by right clicking a file, or group of files, and choosing the script you'd like from the contextual menu (the contents of your script folder may vary from below):



There are four AutoExpand scripts that will be useful to you.
AutoExpand CatID, AutoExpand CatID from Filename, and AutoExpand Filename, and AutoExpand from Category SubCategory

These scripts work in conjunction with the lookup table and category list behind the scenes (_facilitylookups.lua and _categorylist.csv), to break apart the filename, and comparing those pieces to the lookup table, to automatically fill in certain fields in Soundminer.

AutoExpand CatID from Filename takes the filename, which remember is in the following format:

BGPub_BusyInteriorRestaurantBarWallaEvening01_TN_DORY_1024.wav

…and then looks at only the information up to the first _.

It compares this to the _categorylist.csv file, and if it finds a match, it will automatically fill in several fields.

In the example above, it will see the BGPub_, and comparing that to the lookup table, it will set the following fields accordingly:

Category: willl be set to AMBIENCE
SubCategory: will be set to PUBLIC
CategoryFull: will be set to AMBIENCE-PUBLIC
CatID: will be set to BGPub

If the AutoExpand CatID from Filename fails to find a match, it will return an error. This is an indicator that the CatID at the beginning of your file is invalid, and not a match to a designated Category / Subcategory pair. So it works great as a final check before files are committed to your library.

AutoExpand CatID is the lightest of the scripts. It takes the CatID, if present, matches it to the category list, and fills in **Category**, **SubCategory** and **CategoryFull.** It accomplishes the same thing as AutoExpand CatID from Filename but only works once the CatID is already filled in. It does not look to the actual filename to expand.

The AutoExpand Filename script includes the CatID script, but also sets additional fields based on the filename. So again on our example file

BGPub_BusyInteriorRestaurantBarWallaEvening01_TN_DORY_1024.wav

Running it will break the filename apart and fill the following fields:

Category: will be set to AMBIENCE
SubCategory: will be set to PUBLIC
CategoryFull: will be set to AMBIENCE-PUBLIC
CatID: will be set to BGPub
FXName: will be built by auto-splitting on capitalizing (or spaces), then UPPERCASED and become: BUSY INTERIOR RESTAURANT BAR WALLA EVENING 01
Show: will match in the lookup table if it can and be set, in this case Tim Nielsen
Designer: will match in the lookup table if it can and be set, in this case Finding Dory

Either script will error if it cannot find a proper match for the **CatID**. This is to ensure that all files at least have the **Category / SubCategory** set correctly. So for example, if we had called our file:

BgPub_BusyInteriorRestaurantBarWallaEvening01_DSNGER_SHOW_1024.wav or
BGPublc_BusyInteriorRestaurantBarWallaEvening01_DSNGER_SHOW_1024.wav

… either case will fail, because those **CatID**s do not match anything in the _categorylist.csv

The AutoExpand Filename will not throw an error if it cannot find a match for **ShortID** (Designer) or **Source** (Show). In these cases, it will simply leave those fields blank. This is an indicator that the initials you used in the filename don't match the lookup table. The **UserData** optional chunk at the end is simply ignored as well by either script.

You could however fairly easily use a workflow and RegEx or some other method to break out this final part and store in the field of your choosing, for example, to **Notes** or **Microphone**, or somewhere fitting whatever you choose to store in the this chunk.

The AutoExpand Filename WILL throw an error, or get mightily confused, if it cannot successfully break apart the filename into the expected four or five parts. This usually occurs because of extra instances of the _, or lack thereof, which confuse the script. So in the following examples, the script will fail, because it cannot successfully figure out which parts of the filename correspond to its expectation:

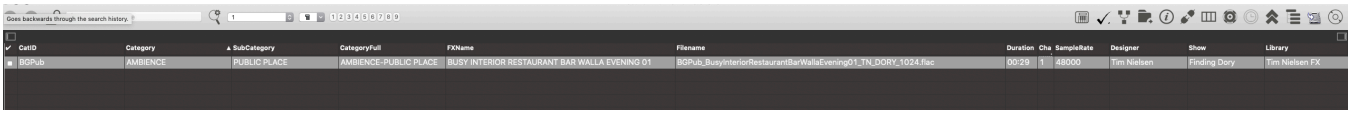
BGPub_Busy_Interior_Restaurant_Bar_Walla_Evening01_DSNGER_SHOW_1024.wav
BGPubBusyInteriorRestaurantBarWallaEvening01DSNGER_SHOW1024.wav
BGPub_BusyInteriorRestaurantBarWallaEvening_01_DSNGER_SHOW_1024.wav

In all of these cases the script will fail. In the second case it will also throw an error because the **CatID** is also invalid (Because it is missing the _ immediately after the **CatID**.

If the script succeeds, it will fill in **Category**, **SubCategory**, **CategoryFull**, **CatID**, **FXName**, **Designer** and **Show**. Of the commonly viewed fields, this would leave only **Description** and Library to be added manually.

So in this case:

Becomes:



The last script, AutoExpand from Category SubCategory, will basically do a reverse lookup to find the **CatID** It will compare Category and SubCategory, and if it finds an exact match, it will fill in the corresponding **CatID.**

Modifying the Category List

The category list is stored in the file called _categorylist.csv, stored in the following path:

~/Library/Application Support/SoundminerV5/_categorylist.csv

On launch, if this file doesn't yet exist, Soundminer will create it with the default Universal Category List that has recently been updated. If this files does exist, but hasn't been modified by the user, SM will update it to the latest version. However if you have gone in and modified the list manually, on launch, Soundminer will detect this and not overwrite the file.

It is a standard .csv (comma separated value) file that can be opened by a variety of programs, including Microsoft Excel (although beware that by default opening it in Excel will not preserve the special characters used in many foreign translations). Apple's PAGES seems to preserve the Unicode characters better than Excel.

The first three columns that must exist, **Category**, **SubCategory** and **CatID**.

Once created, it will look this:

Category	SubCategory	CatID	Explanations	Category_fr	SubCategory_fr	Category_pl	SubCategory_pl
AIR	BLOW	AIRBlow		AIR	SOUFFLE	POWIETRZE	PODMUCH
AIR	BURST	AIRBrst	Sharp air releases	AIR	ÉCLATEMENT	POWIETRZE	WYBUCH
AIR	HISS	AIRHiss		AIR	SIFFLEMENT	POWIETRZE	SYK
AIR	MISC	AIR		AIR	MISC	POWIETRZE	RÓŻNE
AIR	STEAM	AIRStm		AIR	VAPEUR	POWIETRZE	PARA
AIR	SUCTION	AIRSuck	Water pump sucking in water, etc.	AIR	SUCCION	POWIETRZE	SSANIE
AIRCRAFT	BLIMP	AEROBlmp		AVIONS	DIRIGEABLE	LOTNICTWO	STEROWIEC
AIRCRAFT	GLIDER	AEROGlid		AVIONS	PLANEUR	LOTNICTWO	SZYBOWIEC
AIRCRAFT	HELICOPTER	AEROHeli		AVIONS	HÉLICOPTÈRE	LOTNICTWO	HELIKOPTER
AIRCRAFT	JET	AEROJet		AVIONS	À RÉACTION	LOTNICTWO	SAMOLOT ODRZUTOWY
AIRCRAFT	MISC	AERO		AVIONS	MISC	LOTNICTWO	RÓŻNE
AIRCRAFT	PROP	AEROProp		AVIONS	ACCESSOIRE	LOTNICTWO	SAMOLOT ŚMIGŁOWY
AIRCRAFT	ROCKET	AERORckt		AVIONS	FUSÉE	LOTNICTWO	RAKIETA
ALARMS	BUZZER	ALRMBuzr		ALARMES	BUZZER	ALARM	BZYCZEK
ALARMS	KLAXON	ALRMKlax		ALARMES	KLAXONS	ALARM	KLAKSON
ALARMS	MISC	ALRM		ALARMES	MISC	ALARM	RÓŻNE

The first three columns are the most crucial, and must be called Category, SubCategory and CatID (they are case sensitive). These define the ‘Category List’ that Soundminer will use internally for many of the functions of choosing and filling in fields. You should keep them in this order too.

The column called Explanations, because it doesn’t match any field in Soundminer, is simply ignored. It’s use here is to help explain some of the decisions made, and help define some of the more obscure categories. You could also enter your own notes here.

The rest of the columns are the various language translations.

There is also an Excel document available that contains the same information, but with some color coding to make it easier to find information, It looks like this:

Category	SubCategory	CatID	Explanations
AIR	BLOW	AIRBlow	
AIR	BURST	AIRBrst	
AIR	HISS	AIRHiss	
AIR	MISC	AIR	
AIR	STEAM	AIRStm	
AIR	SUCTION	AIRSuck	
AIRCRAFT	BLIMP	AEROBlmp	
AIRCRAFT	GLIDER	AEROGlid	
AIRCRAFT	HELICOPTER	AEROHeli	
AIRCRAFT	JET	AEROJet	
AIRCRAFT	MISC	AERO	
AIRCRAFT	PROP	AEROProp	
AIRCRAFT	ROCKET	AERORckt	
ALARMS	BUZZER	ALRMBuzr	
ALARMS	KLAXON	ALRMKlax	
ALARMS	MISC	ALRM	
AMBIENCE	BIRDSONG	BGBird	Forest Group Birds, etc.
AMBIENCE	CAVE	BGCave	
AMBIENCE	CELEBRATION	BGCele	Parties, New Years Eve, etc.
AMBIENCE	CONSTRUCTION	BGCnst	

If you chose to modify this list, you would simply change the data in the _categorylist.csv file (and if you want in your local copy of the Excel sheet as well if you want).

For example, you might decide that you don't like the UPPERCASE Category and SubCategory style. You could easily TitleCase it, or lowercase it. You could also modify the SubCategory names to be plural if you like, as some people have commented, so ANIMALS-HORSES instead of ANIMALS-HORSE.

What we would emplore you to consider, is to leave the **CatID** and not modify it. This is the potential key to making this list Universal. If that **CatID** exists and everyone agrees to use it, when an incoming file comes in that has that key piece of information, you'll be able to use it to split out to your **Category** and **SubCategory** pair, even if you've changed them.

It is also possible to add columns to this document. If any column's name at the top matched a valid defined field in Soundminer, that information will copy to the matched field when the AutoExpand scripts are run.

For example in the following screen grab, I've added a column called 'Library' and entered "Tim Nielsen FX" into every field in the column.

Category	SubCategory	CatID	Library	Explanations
AIR	BLOW	AIRBlow	Tim Nielsen FX	
AIR	BURST	AIRBrst	Tim Nielsen FX	
AIR	HISS	AIRHiss	Tim Nielsen FX	
AIR	MISC	AIR	Tim Nielsen FX	
AIR	STEAM	AIRStm	Tim Nielsen FX	
AIR	SUCTION	AIRSuck	Tim Nielsen FX	
AIRCRAFT	BLIMP	AEROBlmp	Tim Nielsen FX	
AIRCRAFT	GLIDER	AEROGlid	Tim Nielsen FX	
AIRCRAFT	HELICOPTER	AEROHeli	Tim Nielsen FX	
AIRCRAFT	JET	AEROJet	Tim Nielsen FX	
AIRCRAFT	MISC	AERO	Tim Nielsen FX	
AIRCRAFT	PROP	AEROProp	Tim Nielsen FX	
AIRCRAFT	ROCKET	AERORckt	Tim Nielsen FX	
ALARMS	BUZZER	ALRMBuzr	Tim Nielsen FX	
ALARMS	KLAXON	ALRMKlax	Tim Nielsen FX	
ALARMS	MISC	ALRM	Tim Nielsen FX	
AMBIENCE	BIRDSONG	BGBird	Tim Nielsen FX	Forest Group Birds, etc.
AMBIENCE	CAVE	BGCave	Tim Nielsen FX	
AMBIENCE	CELEBRATIO	BGCele	Tim Nielsen FX	Parties, New Years Eve, etc.
AMBIENCE	CONSTRUCT	BGCnst	Tim Nielsen FX	

The effect of this is this: If any lookup matches a **CatID**, in addition to parsing out **Category** and **SubCategory**, it will also parse whatever other columns match fields. In this case, if there is any match, "Tim Nielsen FX" will be placed in the **Library** field in Soundminer.

Modifying the Facility Lookup

The file _facilitylookups.lua contains the 'Lookup Table'. While categories are defined in the _categorylist.csv, this file is where you will enter custom Designer and Show lookup data. This file is located at the following path:

~/Library/Application Support/SoundminerV5/Scripts/_facilitylookups.lua

This file is a .lua script and at first glance may look confusing. But we are only interested in two distinct chunks right at the top:

```
-- Universal Category Ingestion Script Lookup Table.
-- May 06, 2020. Justin Drury & Tim Nielsen

module(...,package.seeall)
--[[ ================= LOOKUP TABLES ================= ]]--
local lookups={}
-- Designer Field Lookup
lookups.designersLookups={ N1="Name One", N2="Name Two" }
-- Show Lookup
lookups.showLookups={SHOW1="My Show 1", SHOW2="My Show 2" }
```

The –- Designer Field Lookup and –- Show Lookup are where you can define your own matches for the AutoExpand Filename script. In this case, I want to add my own initials, TN and I'll add one show, Finding Dory. So by simply adding these entries into that lines as follows, I will have added these two pieces of information:

```
-- Universal Category Ingestion Script Lookup Table.
-- May 06, 2020. Justin Drury & Tim Nielsen

module(...,package.seeall)
--[[ ================= LOOKUP TABLES ================= ]]--
local lookups={}
-- Designer Field Lookup
lookups.designersLookups={ N1="Name One", N2="Name Two", TN="Tim Nielsen" }
-- Show Lookup
lookups.showLookups={SHOW1="My Show 1", SHOW2="My Show 2", DORY="Finding Dory" }
```

Now when the AutoExpand Filename script runs, when it gets to the **ShortID** Chunk and finds TN there, it will query the lookup table, find a match, and fill in "Tim Nielsen" into the **Designer** field.

Likewise, when it breaks out the fourth part of the filename, the **Source** chunk, it will also query the table, find DORY as a match, and fill in "Finding Dory" into the **Show** field.

Syntax in a .lua script is very important, so take care account of what you're doing in here, be careful of commas and quotes, and follow the example above clearly.