

Readme for SS2_ImportStageData

This script can either import data into an existing SS2 blueprint, or generate one.
Only linear blueprints are supported atm, don't try it on branched ones for now.

Required files

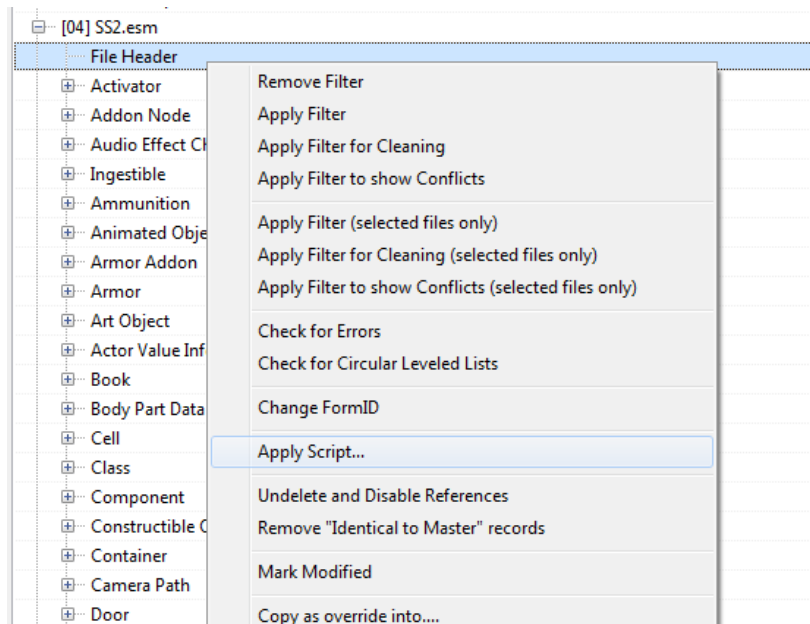
- SS2_ImportStageData.pas
- SS2Lib.pas
- praUtil.pas
- mteFunctions.pas

General Usage

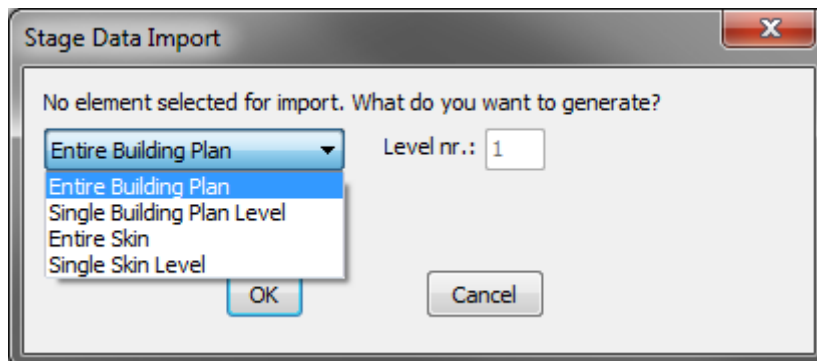
SS2_ImportStageData can be used to either update or generate the following:

- Entire Building Plans
- Single Building Plan levels
- Entire Skins
- Single Skin levels

In order to *generate*, run it on something which is none of these, but is contained in your file. The file's header is a good option:

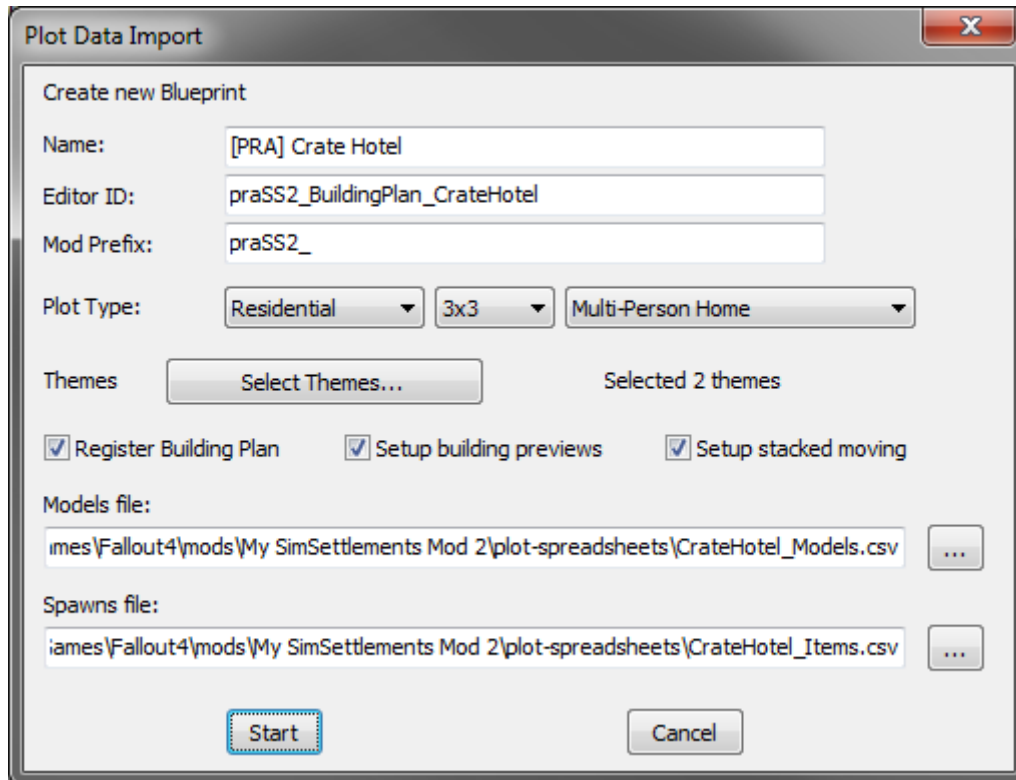


You will be presented with the following menu:



If you want to *update* an existing record, you must run the script on it directly.

Building Plans



The 'Plot Data Import' dialog box is used to create a new blueprint. It contains the following fields and options:

- Create new Blueprint**
- Name:** [PRA] Crate Hotel
- Editor ID:** praSS2_BuildingPlan_CrateHotel
- Mod Prefix:** praSS2_
- Plot Type:** Residential (dropdown), 3x3 (dropdown), Multi-Person Home (dropdown)
- Themes:** Select Themes... button, Selected 2 themes
- ☒ Register Building Plan
- ☒ Setup building previews
- ☒ Setup stacked moving
- Models file:** C:\Fallout4\mods\My SimSettlements Mod 2\plot-spreadsheets\CrateHotel_Models.csv
- Spawns file:** C:\Fallout4\mods\My SimSettlements Mod 2\plot-spreadsheets\CrateHotel_Items.csv
- Start** button
- Cancel** button

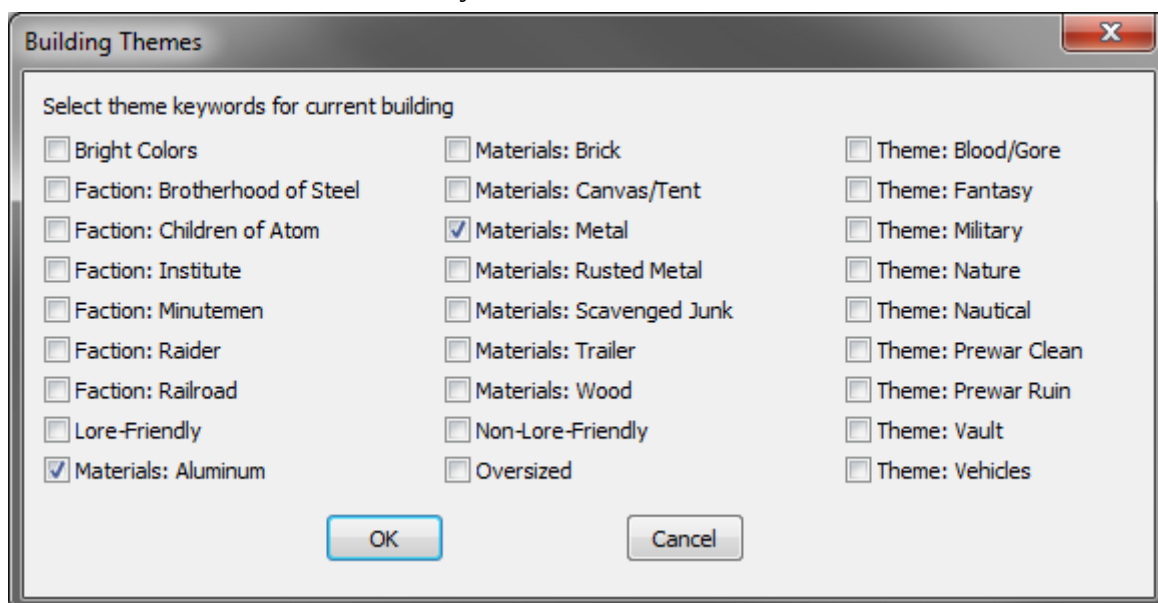
Name: The blueprint's in-game name. Will also be used for the description, you have to change it manually afterwards.

EditorID: Editor ID of the root record, and basis for the ID of other records

Mod Prefix: Prefix to be used for newly-generated records

Plot Type: Main type, Size (including "Interior"), Subtype. The subtype dropdown will change if you change the main type.

Themes: can be used to select theme keywords:



The 'Building Themes' dialog box allows selecting theme keywords for the current building. It contains the following options:

<input type="checkbox"/> Bright Colors	<input type="checkbox"/> Materials: Brick	<input type="checkbox"/> Theme: Blood/Gore
<input type="checkbox"/> Faction: Brotherhood of Steel	<input type="checkbox"/> Materials: Canvas/Tent	<input type="checkbox"/> Theme: Fantasy
<input type="checkbox"/> Faction: Children of Atom	<input checked="" type="checkbox"/> Materials: Metal	<input type="checkbox"/> Theme: Military
<input type="checkbox"/> Faction: Institute	<input type="checkbox"/> Materials: Rusted Metal	<input type="checkbox"/> Theme: Nature
<input type="checkbox"/> Faction: Minutemen	<input type="checkbox"/> Materials: Scavenged Junk	<input type="checkbox"/> Theme: Nautical
<input type="checkbox"/> Faction: Raider	<input type="checkbox"/> Materials: Trailer	<input type="checkbox"/> Theme: Prewar Clean
<input type="checkbox"/> Faction: Railroad	<input type="checkbox"/> Materials: Wood	<input type="checkbox"/> Theme: Prewar Ruin
<input type="checkbox"/> Lore-Friendly	<input type="checkbox"/> Non-Lore-Friendly	<input type="checkbox"/> Theme: Vault
<input checked="" type="checkbox"/> Materials: Aluminum	<input type="checkbox"/> Oversized	<input type="checkbox"/> Theme: Vehicles

OK button **Cancel** button

Register Building Plan: If checked, the blueprint will be automatically registered with your addon quest.

Setup building previews: If checked, the generated blueprint will receive the Level 1 model as a preview.

- Note: it seems that palette indices don't work here, so the previews might have the wrong colors

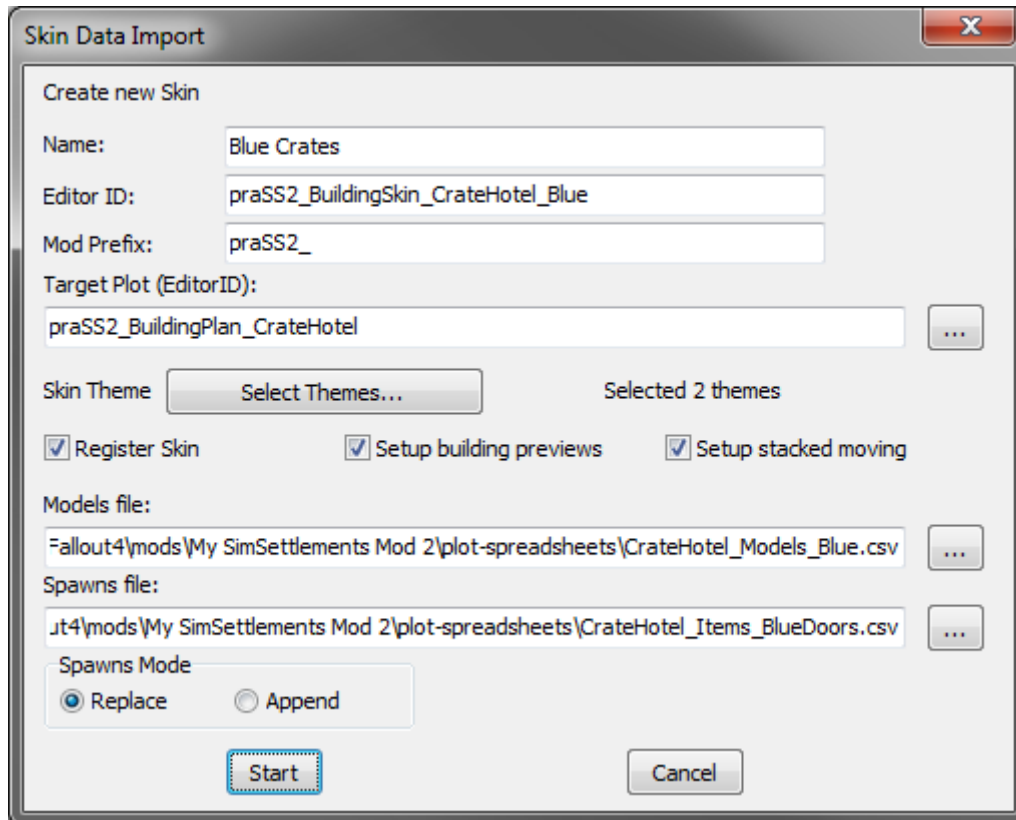
Setup Stacked moving: If checked, the script will automatically create a formlist and a constructible object containing all the building plan's models, enabling them to be moved along with the plot in workshop mode.

Models File: A CSV containing the model data. See below for more information.

Spawns File: A CSV containing the item spawn data. See below for more information.

If you are *generating* a building plan, both the models and the spawns file are required. If you are *updating* one, only one is required.

Skins



Most fields are the same as with the Building Plan dialog. See that section for more information.

Target Plot (EditorID): The editor ID of the target building plan (required). You can write it in directly, or use the [...] button next to it to select one from the list of all plots. Keep in mind that this list takes a few seconds to build.

Spawns Mode: Only relevant if a spawns file is given. “Append” makes the building spawn both it’s normal and the skin’s spawns, “Replace” only the skin’s spawns.

Only one of “Models file” or “Spawns file” is necessary.

Single Levels

The script can also be used to update or generate single levels of building plans or skins. The dialogs look mostly the same, and use the same file formats. However, only the data for the correct level will be used from these files. That is, if you import new data into a Level 2, the importer will disregard Level 1 and Level 3 data in the given file(s).

Generated single levels will be fully standalone, not connected to any parent element.

The Files

The same, or at least, similar, file formats are used for any operation.

The examples shown here are included as SS2_Models_template.csv, SS2_Skin_Models_template.csv and SS2_Items_template.csv

Models

	A	B	C	D	
1	default				
2	MyBP_L1_C1	MyBP_L1_C2	MyBP_L1Final		
3	MyBP_L2_C1	MyBP_L2_C2	MyBP_L2_C2	MyBP_L2Final	
4	MyBP_L3_C1	MyBP_L3_Final			
5					

The first line represents the building materials. The three lines following it stand for the levels 1-3.

The building materials can be the name of a custom static or SCOL, or just the word “default”.

The level rows can have as many entries as you want, but at least one. The final entry is the level’s main model, all entries before it are the construction stages.

In this example, no custom building material model is being used (“default”). Then, Level 1 has three stages (two construction stages before the final), Level 2 has four, and Level 3 only two.

For skins, the building materials entry is ignored. In the following three lines, only the last entry of each line is considered. Thus, a models file for skins can be simplified to this:

	A	
1	Skin Models:	
2	MySkin_L1_Model	
3	MySkin_L2_Model	
4	MySkin_L3_Model	
5		
6		

Spawns

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Form	Pos X	Pos Y	Pos Z	Rot X	Rot Y	Rot Z	Scale	iLevel	iStageNum	iStageEnd	iType	
2	SomeForm	23.5	0.0000	0.0000	0.0000	0.0000	90.0000	1.0000	1				
3	SomeOtherForm	0.0000	10.0000	0.0000	0.0000	0.0000	0.0000	1.0000	2	1	4		
4	YetAnotherForm	0.0000	0.0000	8.0000	45.0000	0.0000	0.0000	0.9500	3				9
5													

Similar to SimSettlements 1, the spawns file is like a CreationKit export, with some additional columns. The first 8 (FormID, Position x, y, z, Rotation X, Y, Z, Scale) are used identically.

- **iLevel** is the building level (1, 2, 3)
- **iStageNum** and **iStageEnd** are now relative to the level and start with 1, not 0!!! Both are optional.
 - If iStageNum is left blank, the object will be spawned at the final stage of the corresponding level.
 - If iStageEnd is left blank, the object will exist for one single stage only.

- These fields are ignored for skins.
- **iType** works as in SimSettlements 1
- **sVendorType** and **iVendorLevel** are relevant for Real Inventory Display only. If your spawned form is a Real Inventory Display Point, you should specify these two.
- **sVendorType** can be either a number 0-6 for a vanilla type, or a custom vendor ID string. If left blank, 0 = General Vendor is assumed.
- **iVendorLevel** is the vendor level, beginning at 0. If left blank, 0 is assumed.
- **iOwnerNumber** is relevant for multi-settler buildings. All objects with an iOwnerNumber of 1 will be given to the first (main) owner, objects with a 2 to the second, and so on. Objects with a 0 (default) will be divided between them.
- **sSpawnName** All blank sSpawnName objects will be spawned, only fill this out to setup conditions where only the first matched with each name will be created at each stage. This allows for having fallback options for a specific spawn depending on circumstances.
- **Requirements** can be set to the Editor ID of an UsageRequirements Misc Item.

In this example, SomeForm will exist at the final stage of Level 1. At Level 2, SomeOtherForm is spawned in at the first Level 2 construction stage, and persists into the fourth stage. At the final stage of Level 3, YetAnotherForm is spawned, but only if clutter is enabled.