Naming conventions
all sections of a name are done in camelCase with no spaces, underscores or special characters. If a number needs to be added, use the format name0X to allow for multiple variations to be generated later. The exact placement of the underscores is important for asset tracking and scripts

The general format of my naming convention is type_package/envName_assetName_extraInfo

Levels:
{Level Type}_{levelName}_{extraInfo}_v{version number, 0XX}

ML_	Master Level	A level used for the organization of the levels under it. Generally this level is kept empty
SL_	Sub Level	The general use sublevel. Anything but enviroment lighting can go here. Practical lighting such as a torch with a point light can. Generally this level does not include shot specific geomotry or characters.
LSL_	Lighting Sub Level	Lighting for the overall look of the scene. Includes skylight directional lights, post process, skies, fog and any other lighting that is not practical (though if you need to turn it on and off in 2 looks, place it here)
AL_	Animation/sequence level	Shot specific geometry, characters and lighting

You can have as many sub levels in a scene as you want (I have used upwards of 60). They are useful for organization as you can quickly toggle visibility on all objects in a sub level. It also means one person can check out the geo sublevel while someone else has the lighting sub level checked out and both of their work will be saved.

Textures:
TX_{package/envName}_{materialName}_{code}

_col	Base color, albedo, diffuse, etc	Set to pack without alpha
_orm	Occlusion Roughness Metallic packed texture	Set to linear color, no srgb, mask
_nrm	Normal	Set to world normal
_opa	Opacity (usually separated from the base color)	Set to linear color, mask no srgb BW
_emm	Emissive	
_msk	Mask	
_rgh	Roughness (usually baked into an ORM)	
_ao	Ambient Occlusion (usually baked into an ORM)	
_met	Metallic (usually baked into an ORM)	
_disp	Displacement	

_hgt	Height	
_flm	Flowmap	
_lmp	Lightmap (fake)	
_imp	Imperfection/Noise	
_subc	Subsurface color map	
_subi	Subsurface intensity map (mask)	

Textures must be power of 2. Resolutions must be made of 512, 1024, 2048, 4096, 8192. Rectangular textures are fine so long as they are 1024*2048 or similar

Materials:
{MaterialType}_{package/envName}_{materialName}

MM	Master Material	What UE4 calls materials. They operate like shaders in other programs and use node graphs. Generally you do not apply master materials to objects in a scene, instead use a material instance
MI	Material Instance	UE4 calls them material instances. If your master material has parameters set up, the instance will show easy to use sliders and value inputs instead of a node graph. Using one master material and 10 material instances of it is significantly better performing than a full master material on each object
MMLF	Light Function Master Material	UE4 material for light functions (gobos, cookies)
MLFI	Light Function Material instance	UE4 material instance for light functions (gobos, cookies)
MF	Material function	A subset of a material graph used to organize and reuse code.
SSP	Subsurface profile	A gradient color map for light scattering. Useful for Jade, skin and other subsurface materials. Requires a material set up to use them.
DMI_	Decal Material Instance	The material instance is a decal. It works like a 2d texture projector onto any surface and does not require an associated mesh

Static Meshes:
SM_{package/envName}_{optionalGroupName}_{meshName}

Skeletal Meshes
SK_{package/envName}_{optionalGroupName}_{meshName}
SK_{package/envName}_{optionalGroupName}_{skeletalMeshName}_Anim
SK_{package/envName}_{optionalGroupName}_{skeletalMeshName}_PhysicsAsset
SK_{package/envName}_{optionalGroupName}_{skeletalMeshName}_Skeleton

Other:
LUT_{package/envName}_{LUTname}
Thumb_{package/envName} Thumbnail