Object Attributes and Semi-Attributes

All objects (beds, rugs, counters, Sims, job data, controllers, etc) have attributes – general-purpose data holders (aka variables) that can be used to store information about the object. For example, who the driving licence belongs to and how many penalty points they have on it.

If an object has an associated GLOB resource, that is, it has an associated semi-globals group, it can also have semi-attributes - again general-purpose data holders.

There is really no difference between attributes and semi-attributes - they both hold general-purpose values. Attributes tend to be used by "local" code, while semi-attributes tend to be used by "semi-global code" - although there is nothing stopping "local code" from accessing semi-attributes nor "semi-global code" from accessing attributes.

Each object has its own set of attributes and semi-attributes - two different driving licence cards are owned by different Sims and have different penalty points values.

Attributes

Every object has at least eight attributes, depending on the setting of the 0x003A "num attributes" entry in the object's OBJD resource (found under "09. Data Space" in SimPe). For reasons known only to Maxis, values of 0 to 8 in this entry are interpreted as 8, values of 9 or greater are interpreted as the given value, that is

“num attributes” value	Available attributes
0	8
1	8
2	8
3	8
4	8
5	8
6	8
7	8
8	8
9	9
10	10
11	11
etc	

Attributes can (and should) be assigned labels by adding entries to the 0x0100 STR# resource in the same group (usually 0xFFFFFFFF) as the object.

Because all objects have at least 8 attributes and they don't have to have labels, when adding a new attribute to an object (you don't have in-depth knowledge of) it is safest to assume the first 8 attributes are in use. If "num attributes" is 8 or greater, add how many attributes you need to it. If "num attributes" is less than 8, assume it is 8 and add how many attributes you need. Don’t forget to give your new attributes labels!

Semi-Attributes

Semi-attributes are different - an object has as many semi-attributes as there are entries in the semi-global group's 0x0100 STR# resource. To add a new semi-attribute, over-ride the semi-global's 0x0100 STR# resource and add new entries to the end of the list.

Note: The original Hack Conflict Detection Utility (HCDU) does NOT detect STR# resource conflicts - so you could have two mods that both override a semi-global group's STR# 0x0100 resource, and it WON'T report a conflict. My HCDU Plus utility does detect STR# resource conflicts.

Persistence

Persistence: In computing, the process of saving data associated with something.

Attributes and semi-attributes store values that need to be "remembered" by the associated object - for instance if a light is on or off, the bug state of a plant, the quality of a fish, the progress on fixing up a car, etc. When you resume play of a lot, the light is still on, the tree still needs to be sprayed, the fish will make a "sparkly" meal, the car still won't run etc.

Non-Residential Lot Persistence Issue

Objects on non-residential (community, secret, hobby, etc) lots behave differently to those on residential (home, apartment, dorms, etc) lots. They do NOT remember (persist) their attributes and semi-attributes, so do NOT remember how they were left (light on or off, etc) from the previous trip there - but revert to how they were when the lot was built.

If an object on a non-residential lot needs to remember something (for example, my Info Card remembers what skill a Sim reading it increases), you will need to store that information into a token in the Lot Inventory - difficult, but not impossible, but beyond the scope of these notes!

=== END ===