```
________________________________________________
THE MORTAR & PESTLE-OLOGY OF MULTI-TILE OBJECTS

        Every tile in an IFF file has its own individual OBJD resource. If your
        toy is a single-tile OBJD, there will only be one OBJD resource in the IFF
        with a few exceptions to do with blanked-out reserve tiles and dynamic
sprites.

        If you're working with a multi-tile object, there will be an OBJD for each
        tile plus an additional OBJD called the master object, which doesn't show
up
        in the game, see more on that below. Also in some cases to do with dynamic
        sprites and/or reserved tiles, some multi-tile objects may have more than
        one OBJD per tile plus the OBJD for the master ID. I'm including an example
        herein to show how that works as well.

MASTER ID
        Every multi-tile object has a master object, which doesn't show up in the
game,
        but keeps the subtiles together. The Master ID is a unique number (in the
file)
        that The Sims can use to find out which objects are part of one multi-tile
object
        or another.

SUB INDEX
        The tile which the OBJD applies to. The decimal value for a single-tile
object
        must be set to 0 for the object to work. If the object is a multi-tile
master object
        (which doesn't show up in the game), the decimal value must be set to -1 .
The formula
        for the decimal value for tiles in a multi-tile object is row*256+col. Rows
and columns
        start at 0. Sub index value 0 is closest to the screen, while rows go left,
and columns
        go right from that point. Objects may not function properly if tiles aren't
all adjacent.

Maxis wrote the paragraphs above, so don't blame me if you can't make anything of
what they're saying. Since the Sub index part is jabbergeeky, I'll give you some
examples to demonstrate what it means since I know it all. I may just redo TMog
myself in fact if I have an extra weekend and Don never shows up again. j/k
<written about a month after I lost contact with Sarge in January of 2003>

--------------------------------------------------------
Everything You Ever Need To Know About The Master Index
and Wish You Never Needed To Know Even This Much
--------------------------------------------------------
The Master Index will usually be 1 if there is only a single toy in the
IFF file. When Maxis packs 2 or more toys in the same IFF file as they do with
things of a kind such as sofas, chairs, tables, refrigerators to name but a
few, that's where the Master Index comes into play. Theoretically, if there's only
a single toy in an IFF file, the Master Index could be any number just so long as
it's the same for all tiles in the IFF file. TMog is apparently programmed to
set the Master Index to 1 when we clone a single toy out of an IFF file with
several different toys. If there are left-over draw groups from other toys inside
your IFF file, it will be imperative that you make sure your toy's Master Index is
pointing to the correct number, which may or may not be 1. See painting4.iff below
for a stupid example of a single toy with a Master Index higher than 1 and why
it's stupid.

--------------------------------------------------------
The Sub Index Number
--------------------------------------------------------
```

This file contains a few case examples of objects and their Sub index numbers for each tile with explanations of how maxis arrived at the index numbers. If you don't already understand the formula, you will after you glance through these examples.

Keep the formula in mind while you look at the examples...

> "The formula for the decimal value for tiles in a multi-tile object is row*256+col. Rows and columns start at 0. Sub index value 0 is closest to the screen, while rows go left, and columns go right from that point."

What they mean by "closest to the screen" is that tile which is literally closest to you
when you are viewing the complete set of tiles in the SE rotation. SE is the rotation that
TMog defaults to when you open the viewscreen, i.e., the 3rd notch position. I've never yet
been able to figure out what planet you'd have to be on for that to be SE from anyone's
perspective, but forget I even mention it. Because you won't ever have a problem with this
if you just add this parenthetical to Maxis' formula paragraph after the words "closest to
the screen"...

> "when viewing the toy in TMog from the viewscreen's default rotation"

This makes sense, come to think of it. That 3rd or SE rotation is the reference point,
and that would explain why it is in fact the default rotation in TMog instead of defaulting
to, for instance, the first notch position. It obviously has some kind of logical application
in the maxis geekspeak scheme of things. And this is about as close as we're probably ever
gonna get to that logic.

It will help if you open TMog and look at the files example'd below while you read my
notes. It's the only way you'll be able to make sense of the part that states "rows go
left, and columns go right." Once you understand the process, it's a piece-a-cake, I'm
sure you'll agree.

```
------------------------------------------------------------
1-Tile Objects
(applies to all 1-tiles, except those w/separate sprites for
animations & a very few others with reserved tiles)
------------------------------------------------------------
Master ID       0
Sub index       0

-----------------------------
2-Tile Object (3 OBJD's)
Case Ex: SF_Fridge2Tile.iff
-----------------------------
Objd#           16807 SF_Fridge Tile 1
Master ID       1
Sub index       0

Objd#           16808 SF_Fridge Tile 2
Master ID       1
Sub index       256

Objd#           16809 SF_Fridge 2Tile
Master ID       1
```

```
Sub index       -1


16809 is the Master Object, the one that doesn't show up in the game.

[Image 1] 16807, Tile 1 is the aqua fridge tile "closest to the screen",
it sits at row 0, col 0. 0 X 256 + 0 = 0. Ergo, its Sub index is 0.

[Image 2] 16808, Tile 2 is the white fridge tile. "Rows go left, and columns
go right" per the formula, so it sits at row 1, col 0. 1 X 256 + 0 = 256.


----------------------------
2-Tile Object (5 OBJD's)
Case Ex: McFoodCart.iff
----------------------------
Objd#           16807 - Food Counter - McDonalds
Master ID       1
Sub index       -1

Objd#           16808 - Food Counter - McDonalds - A
Master ID       1
Sub index       0

Objd#           16809 - Food Counter - McDonalds - B
Master ID       1
Sub index       256

Objd#           16821 - Food Counter - McDonalds - Reserve A
Master ID       1
Sub index       1

Objd#           16822 - Food Counter - McDonalds - Reserve B
Master ID       1
Sub index       257

This works exactly as the previous example, except there are 2 additional
OBJD's in this file. The Base Graphic ID #'s of reserve tiles are always
set to 0, which means there are no draw groups and/or sprites for reserve
tiles. Explaining the purpose of reserve tiles is beyond the scope of this
tutorial. My advice is don't be building any bases that require them if you
don't absolutely have to. Particularly if you don't know what they're for. ha

The game treats their Sub index numbers as though they were 3rd and 4th
tiles. See example of 4-tile dining table below.

----------------------------------------------------
3-Tile Object (4 OBJD's)
Case Ex: painting4.iff
(3-tile world map painting that shipped with Hot Date)
----------------------------------------------------
Objd#           16840 - Painting - World Map
Master ID       9
Sub index       -1

Objd#           16841 - Painting - World Map - Tile 1
Master ID       9
Sub index       0

Objd#           16842 - Painting - World Map - Tile 2
Master ID       9
Sub index       256

Objd#           16843 - Painting - World Map - Tile 3
Master ID       9
Sub index       512
```

```
16816 is the Master Object, the one that doesn't show up in the game.

Tile 1 is "closest to the screen" and sits at row 0, col 0. Ergo, its Sub index =
0.
Tile 2 is the middle tile. "Rows go left", so it sits at row 1, col 0. 1 X 256 + 0
= 256
Tile 3 is the remaining tile, and it is left of the previous 2, so it sits at row
2, col 0.
2 X 256 + 0 = 512.

There is only 1 toy in this case example IFF file. However, as is often the case
with
maxis' IFF files, they left all the draw groups from all the hot date paintings in
this
file when they broke it off to make painting4.iff. This is why the Master ID number
is 9.
Perfect example of the kind of fluff they leave in the files, one of the things we
talked
about a fortphonenight ago. No wonder the feckin game is so over-taxed on
resources.
Sloppy, sloppy, sloppy.

----------------------------------------------
4-Tile Object (5 OBJD's)
Case Ex: Moderate Dining Table from base game
(the wooden jobbie with white stair-post legs)
----------------------------------------------
Objd#           16816 Dining Table - Moderate
Master ID       1
Sub index       -1

Objd#           16817 Dining Table - Moderate - A
Master ID       1
Sub index       0

Objd#           16818 Dining Table - Moderate - B
Master ID       1
Sub index       256

Objd#           16819 Dining Table - Moderate - C
Master ID       1
Sub index       1

Objd#           16820 Dining Table - Moderate - D
Master ID       1
Sub index       257


16816 is the Master Object, the one that doesn't show up in the game.

Tile A is "closest to the screen", so its Sub index value is 0. (row 0, col 0)
Tile B is at row 1, col 0. 1 X 256 + 0 = 256.
Tile C is at row 0, col 1. 0 X 256 + 1 = 1.
Tile D is at row 1, col 1. 1 X 256 + 1 = 257.


----------------------------------------------
9-Tile Object (10 OBJD's)
Case Ex: Don's BigOMama Teapot-in-a-Tempest
----------------------------------------------

Objd#           16807 - base - 3x3
Master ID       1
Sub index       -1
```

```
Objd#          16808 - base - 3x3 - frame 00 tile 00
Master ID      1
Sub index      0

Objd#          16809 - base - 3x3 - frame 00 tile 01
Master ID      1
Sub index      1

Objd#          16810 - base - 3x3 - frame 00 tile 02
Master ID      1
Sub index      2

Objd#          16811 - base - 3x3 - frame 00 tile 03
Master ID      1
Sub index      256

Objd#          16812 - base - 3x3 - frame 00 tile 04
Master ID      1
Sub index      257

Objd#          16813 - base - 3x3 - frame 00 tile 05
Master ID      1
Sub index      258

Objd#          16814 - base - 3x3 - frame 00 tile 06
Master ID      1
Sub index      512

Objd#          16815 - base - 3x3 - frame 00 tile 07
Master ID      1
Sub index      513

Objd#          16816 - base - 3x3 - frame 00 tile 08
Master ID      1
Sub index      514
```

NOTES on Teresa's problem with her 3-tile test carpet
and what might be causing it

Open TMog and view Don's teapot. Leave the grid on, and be sure not to move the rotation when the "View Object" screen opens, leave it set to the default SE rotation (3rd notch position). In the Multi Tile Object: drop-down, highlight the first tile below "All Tiles", i.e., the one named...

"base - 3x3 - frame 00 tile 00"

You'll see the tile highlighted in the view screen, this tile's Sub Index number is 0, as should be the case. Notice that it is the tile which is "closest to the screen."

Use your down arrow to cursor through the rest of the 8 tiles, and as you do so, you'll see how perfectly the frames move right to left in the first row, then the succession jumps to the 1st (left) column of the 2nd row and continues left to right, then jumps back to the left column of the 3rd row and continues again left to right. The sequence is perfect and totally visually intuitive.

Now go look at most maxis objects in the game, particularly those made after the base game, select the first tile and cursor through the sprites. There is no visual methodology whatsoever to at least 90% of the rest of the objects in the game (including custom objects, of course, since just about all custom objects are cloned from maxis ojbects). It's a mass of confusion.

This stems at least partly from a lack of internal controls early on at Maxis. This is not conjecture on my part, it's something I have proof of in that Don has given me copies of many internal documents from maxis, including "post mortems" describing minor and major failures, most of which were generated before Don left the company's employee.

Several documents include sections on positioning of objects and meshes in the max viewports. These protocols were obviously shrugged off by people working in the art department, and either ignored or unknown to human resource graphics houses retained by maxis off-premises. It's very disheartening to read some of the narratives, so we're not going to share it with you guys. Besides which you've got too much else to do. And Heather will first have to learn to read her personal email before she gets a chance to read the juicy loosy. hahah

The point is this... Even though it's very difficult to follow any apparent methodology as to sub-index sequencing, the reason that multi-tiles do stick together at all is because the formula was adhered to. Now we enter the Simprovian NoowAge where custom content makers can finally build new objects from scratch if they're very inventive or have the Mogman on their team.

```
-----------------------------
Below is a small extract from a maxis internal document entitled CONTENT CREATION
RULES.
-----------------------------
3.10 Modeling
        Because of the prerendered nature of the sprites in The Sims, there are few
real restrictions
        on modeling other than what looks good in the render. However, for means of
registration and
        correct lighting there are some rules.

3.0.11 The Cage
        All objects must be placed within the cage, which is a space enclosed by
the (x,y) world
        coordinates (1.5,1.5),(1.5,-1.5),(-1.5,-1.5),(-1.5,1.5), defining a square
3'ï¿½on a side.

3.0.12 Orientation
        All objects must be placed so that their front is showing in the Front Max
viewport (facing
        down the negative y world axis). This is checked by making sure that the
1st frame of render
        shows the object facing away and to the right.
------------------------------
```

The part that is important to take note of is the paragraph entitled "Orientation". Unbelievably (to me, at least), what seems to have happened early on with toys is that the various artists employed both internally and through outside processors did not follow this very basic and seemingly obvious rule. I imagine this came about at least partly due to confusion over things whose "front" was more arbitrary and less obvious than, say, a sofa or kitchen counter. Take rugs, for instance. If someone asked you "Which is the front side?" of a carpet in your home, how would you answer them? You'd probably point to the TOP of the rug. But if they wanted to know which was the front side of the 3rd dimension, unless it was a very obvious one-way pattern or scene, you would probably do the simsey shrug in response.

The reason I believe this is what happened is because a few earlier objects are positioned in TMog's view screen differently from later versions of the same object type. Rugs, to use the example again, are cuckoo. You'll start to see what I'm talking about if you open various objects and look at them in TMog (in the SE default rotation) to assess whether or not you would say they're facing frontward.

For example, look at any painting. When TMog's Viewscreen opens, you'll see that paintings are positioned SE as though they are facing frontward into the camera's POV.

Now look at all the beds. Uhoh, they're all unmistakably facing backward away from the camera's POV.

Look at the African Fountain (fountain3x3deluxe2). It would appear to be facing frontward seeing as how the elephant's trunk and head are forward looking into the camera's POV.

Now look at the Superstar Fountain (fountain3x3superstar). The center horse element has definitely got his back to the camera.

The Large Downtown Fountain that shipped with Hot Date (fountainlargehd) is also positioned with its back to the camera.

So much for Internal Controls, Protocols and Creation Rules. As you've heard me say over and over, the video game manufacturing industry could take a hint from film studios and understand what a Continuity Department is all about. Without kickass continuity experts, the movie industry would barely be into the early talkie phase even at this point in history. [Note to Don: Something you might want to ask Will if any of his companies have ever considered looking into].

Teresa, if you'll send me your test 3-tile rug that isn't lining up, I can take a look at it and tell you if the sub-indices are causing the problem. I would have to know how it sits in the SE rotation before knowing for sure if this is the problem. However, if you want to use an ascii-stick-figure illustration to see if you can test the IDs yourself, here's something that might help without your having to send me a copy. A 3-tile rug would sit like this...

```
 /
```

...or like this...

```
 \
```

Either way, the bottom third of either slash would be the tile whose sub index number would be 0.

If it sits like the first slash, the middle tile's sub index should be 0, and its remaining top tile sub index should be 2.

If it sits like the bottom slash, the middle tile's sub index number should be 256, and its remaining top tile sub index should be 512.

If this doesn't help or you're still not sure, send me the rug that's all gimpy as I might get to it before Don checks his email, particularly if you're up and send it in the next couple hours or so.

If it turns out to be a sub-index id issue, Don will be able to fix this tout swit with no problem. I know for positive sure that Don understands the formula, well duh. However, since he was not a content creator, he himself may not have put some of this information together about why the drawgroups and sprite-sets often seem so non-sensical.

I would be interested to know how the sub-index id's were assigned at maxis and by whom if Don can shed any light on that. I have access to a document where there was a decision made to have the artists do more coding details, this might be be a perfect example of why that was a bad decision. haha

Sub Index numbers are not anything you (Teresa and Heather) have ever had to consider since you clone from TMog, work in Max, use the plug-in to export to TGA, then do your post work in 2D and resend the sprites to TMog based on the cloned sprites. There would be no occasion for you to ever deal with sub-index numbers unless you were enlarging a base, am I correct? Same with me except for a few occasions when I have enlarged a few bases. Which is why I ever wrote the original of this tutorial in the first place. I was building a lemonade stand base for Heather and used a 1-tile bar to work from. That is the genesis of this document. Yes, I do have that base, Heather. But never mind for now, it's not weddingly thematic. And Don will vastly improve upon it and turn it into lemon simoleons later on of course. We're saving it for the Kiddy Rompers CD. :\

Now that original bases are a done deal, I'm making it a hard and fast rule that everybody get in synch on the Formula herein-discussed (Don will be the final arbiter on the rule). I also believe it would behoove us to work toward developing a conformed policy regarding positioning of objects as much as that is feasible to institute. Take the example of Don's perfectly sequenced Teapot. It is facing with its back to the camera in the default SE rotation. Don was merely the brains of the game, of course, not a pixelslave, so I'm thinking some of this will be newsworthy for him (particularly seeing as how his almost perfect teapot is facing away from the camera in defiance of near perfection, hehe). We'll need Don's input on this before making any final decisions, but in my thinking, I believe objects should face front-left in the SE rotation *as much as is feasible without drastically impacting anyone's time to conform.* For example, I hardly expect you guys to rewrite your massive bed templates and actions just so the rest of your beds will face leftwardly forward in the SE rotation unlike any of our beds are positioned to date. I'm sure you both concede my point.

I'm including a copy of a file I made several years ago that I still have to refer to constantly to remember how rotations should face. I gave Dariene a copy a few years ago, and she still says she always has it on her desktop anytime she's building any kind of toy. It wasn't until some time after I made this file that I realized many of the maxis objects do not conform to a consistent view basis. However, this file reflects the positioning of the vast majority of toys, using the base game 1-tile shower as the model/example. It should be self-explanatory when you look at it. But whatever you do, don't even try to figure out the logic of the naming of the directions. Don and I have already smoked a doobie over this one while pondering whassup wittit and other mysteries of the corporate llama mindset. Let it go, you ain't gonna figure it out, trust me on this one. ;)

```
------------------------------------------------
Version Edit: 30/Apr/2004 1:36am pactime -dke
------------------------------------------------
```