HLU and OSMM merge: a brief guide to the code

These rough notes describe how to use the set of python scripts developed by Alison Smith and Martin Besnier at the Environmental Change Institute, University of Oxford, for creating a habitat base map suitable for natural capital mapping. The scripts are research tools and have not been rigorously tested or documented.

The tools are designed to work with OS Mastermap (OSMM) as the base map. They then merge in a Phase 1 habitat map. The code is set up to work with the Phase 1 habitat and land use (HLU) map for Oxfordshire, provided under license from the Thames Valley Record Centre (TVERC). However, it should be possible to use any similar datasets by modifying the input parameters for the main merge script, and possibly modifying the pre- and post-processing steps to work with the habitat classifications in the alternative datasets (if they differ).

The most straightforward way to merge together two or more habitat datasets would be though a series of Intersect or Identity operations in ArcGIS. However, the habitat datasets we work with tend to have boundaries that do not exactly match OS Mastermap boundaries. Therefore this method, when performed at county scale, creates tens of thousands of tiny ‘slivers’ at polygon edges, which can cause problems when attempting to perform subsequent geoprocessing operations.

The novel aspect of the code (designed originally by Martin Besnier, a visiting researcher from the Université Paris Sud) is that it merges two polygon datasets (a base map and a set of new features), while avoiding the creation of an excessive number of slivers due to boundary mismatch. It does this through identifying polygons in the base map that need to be split to match the boundaries of the new features, while ignoring minor differences in the boundaries due to inaccurate mapping (which would generate slivers). The output is faithful to the base map boundaries as far as possible, though minor differences (just a few cm) may arise during one of the sliver elimination steps.

A sequence of scripts have been developed which:

1. pre-process the OSMM and habitat input maps (e.g. to remove duplicate and overlapping features);
2. merge the two polygon datasets together without generating too many slivers;
3. generate a unified ‘Interpreted habitat’ attribute field that uses a set of rules to identify the habitat based on information from both OSMM and the habitat map;

These three scripts, plus a ‘MyFunctions’ script containing frequently used custom-built functions, form the core of the habitat base map generation. Further scripts (not described in this document) have also been developed that merge in Agricultural Land Class, fifteen different habitat designation datasets, OS Mastermap Greenspace and OS Open Greenspace, and public access information, to create a more detailed natural capital base map.

# Instructions for running the code

1. All the scripts should be in the same folder as the MyFunctions.py script, which supplies useful common functions. For each stage of the process, edit the script to set the workspace directory and all the other parameters, and run the code. Check the outputs carefully.
2. The code is set up to work for two main options:
   1. Phase 1 habitat data is available
   2. Phase 1 habitat data not available – use CEH Land Cover Map and Natural England Priority Habitat data instead.

We currently run the scripts for Oxfordshire first, using Phase 1 data, and then run for the rest of the Arc using LCM / PHI data.

1. Download the geodatabase version of OSMM topographic area from Edina Digimap or OS (assuming you have a license). Do not use a shapefile version, as field names get truncated and code does not work.

## Using Phase 1 habitat data

1. Check habitat data (e.g. Oxfordshire Phase 1 HLU) for ‘unidentified’ habitats, and use BAP or other info to determine what they are – otherwise delete them (OSMM definition will be used instead). Check for incorrect spellings in Phase1habitat field (duplicates with upper and lower case can be left in, as they should be resolved by the code).
2. Open the **Merge\_OSMM\_HLU\_ Preprocess.py** script:

* Set the workspace gdb path in the code (NB - it is faster to run this on your hard drive rather than a network server). The code currently expects a geodatabase called Merge\_OSMM\_HLU.gdb.
* Depending on what steps you want to run, change the flags in the parameter section to True or False. If the files have not been clipped to the exact boundaries, set ‘clip\_to\_boundary’ to True, name the input files HLU\_in and OSMM\_in, add boundary file to the gdb and set the name of the boundary feature class (e.g. “Oxfordshire”) as a parameter in the code. Otherwise the code expects the input files to be named HLU and OSMM (this can be changed in the parameter section).
* Run the script. This erases manmade features and water from HLU, removes overlaps and eliminates slivers, checks and repairs geometry; removes overlapping features (landforms and pylons) from OSMM; and removes unnecessary fields (enter the list of habitat fields to keep in the script: currently POLY\_ID, PHASE1HABI, BAP\_HABITA, "BAP\_HABI00", "SITEREF", "COPYRIGHT", "VERSION").

1. **Merge\_into\_Base\_MapV5a.py:** open the script, set the merge\_type parameter to “OSMM\_HLU”, check the workspace and all the other parameters and run the code. This can take about 14 hours for merging HLU habitat data into OSMM for the whole of Oxfordshire.
2. **OSMM\_HLU\_Interpret.py**: set ‘region’ to ‘Oxon’. This adds the habitat interpretations, by combining OSMM and Phase 1 appropriately. Note: If OSMM is "undefined" this usually means the area is under development or scheduled for development. The ‘undefined\_or\_original’ flag allows the user to choose whether to map these areas as "undefined" or as the current / original habitat pre-development.

## Using CEH Land cover map and NE PHI data

(These scripts not yet uploaded to GitHub but can be on request)

1. All scripts run in a loop through LADs (leaving out Oxfordshire LADs as we do that separately).
2. **Prep\_OSMM.py** – copies tiles from download folders into a single folder, merges into a single feature class in a geodatabase, and clips to LAD boundaries.
3. **Setup\_LAD\_gdbs.py**. Has several stages – set any that are not needed to ‘False’. At present it is set to ignore Oxfordshire as we have a separate procedure for generating the Oxfordshire data, using Phase 1 habitat.
   1. Prepare PHI data. There are three separate datasets – the main PHI data, plus Wood Pasture and Parkland (WPP) and Open Mosaic Habitats on previously developed land (OMHD). For all three datasets, dissolve on habitat field (Main\_habit or PRIHABTXT), convert to single part, delete polygons <10m2, and copy habitat field to new field called ‘PHI’, ‘WPP’ or ‘OMHD’. Then union all three datasets. WPP and OHMD are not very accurate, e.g. WPP maps whole parkland areas with a mix of habitats including grass, fields, woods, buildings and plantations, and often overlaps with other PHIs. So let OSMM woodland, water, manmade and other PHI habitats take priority.
   2. Set up individual geodatabases for each LAD:
      1. Create new gdb
      2. Copy in the OSMM data for this LAD
      3. Create boundary feature
      4. Clip Arc-wide LCM and PHI data to the boundaries of the LAD, and copy in.
4. **OSMM\_HLU\_Interpret.py** – set ‘region’ to ‘Arc’, and ‘simplify\_OSMM’ to True but other stages to False. This will interpret a simplified habitat from the OSMM Make, Descriptive Group and Descriptive Term.
5. **METHOD 1 (intersect PHI). Run Arc\_LCM\_PHI.py**  - combines with CEH land cover map and Natural England priority habitat data. Set ‘merge\_or\_intersect’ = “intersect”. There are two main stages:
   1. Process\_LCM: Add data from CEH Landcover Map 2015 by defining agricultural land or ‘general surface natural’ as either arable or improved grassland. Then create a joint ‘Interpreted habitat’ field that assigns either the OSMM habitat or the LCM habitat.
   2. Process\_PHI: delete landforms from OSMM data, intersect (using ‘Identity’) with PHI data, then interpret PHI by copying to Interpreted Habitat but only when OSMM is not manmade, garden or water. For WPP and OMHD, keep fine detail of other habitats and only copy these designations across for generic habitats (e.g. farmland). PHI boundaries match OSMM quite well, though with some genuine split polygons, so the intersect (Identity) doesn not create too many slivers, e.g. Aylesbury goes from 455 single part slivers (<1m2) for OSMM\_LCM to 2372 after intersect with PHI, which is not too bad out of approx. 300,000 polygons. However, if time, it is neater to use the alternative method (step 6).
6. **METHOD 2 (merge PHI) ALTERNATIVE to step 5:** Merge rather than intersecting PHI, to reduce slivers. For example, this only increases slivers from 455 to 457 (single part) for Aylesbury. There are relatively few PHI polygons so the Merge does not take too long – about 50 minutes per LAD, 12 hours for all LADs (except Oxfordshire). To do this,
   1. run **Arc\_LCM\_PHI.py** with ‘merge\_or\_intersect’ = “merge” and ‘step = 1’. This should set ‘intersect\_PHI’ and ‘interpret\_PHI’ to False but all other steps to True,
   2. run **Merge\_into\_Base\_Map.py** with merge-type set to ‘Arc\_LCM\_PHI’,
   3. **re-run Arc\_LCM\_PHI.py** with ‘step = 2’. This should set ‘process\_PHI’ and ‘interpret\_PHI’ to True and all other stages to False, to copy the correct habitat interpretation across from the PHI, WPP and OMHD fields to the Interpreted\_habitat field.

# Merging OSMM and HLU – method notes

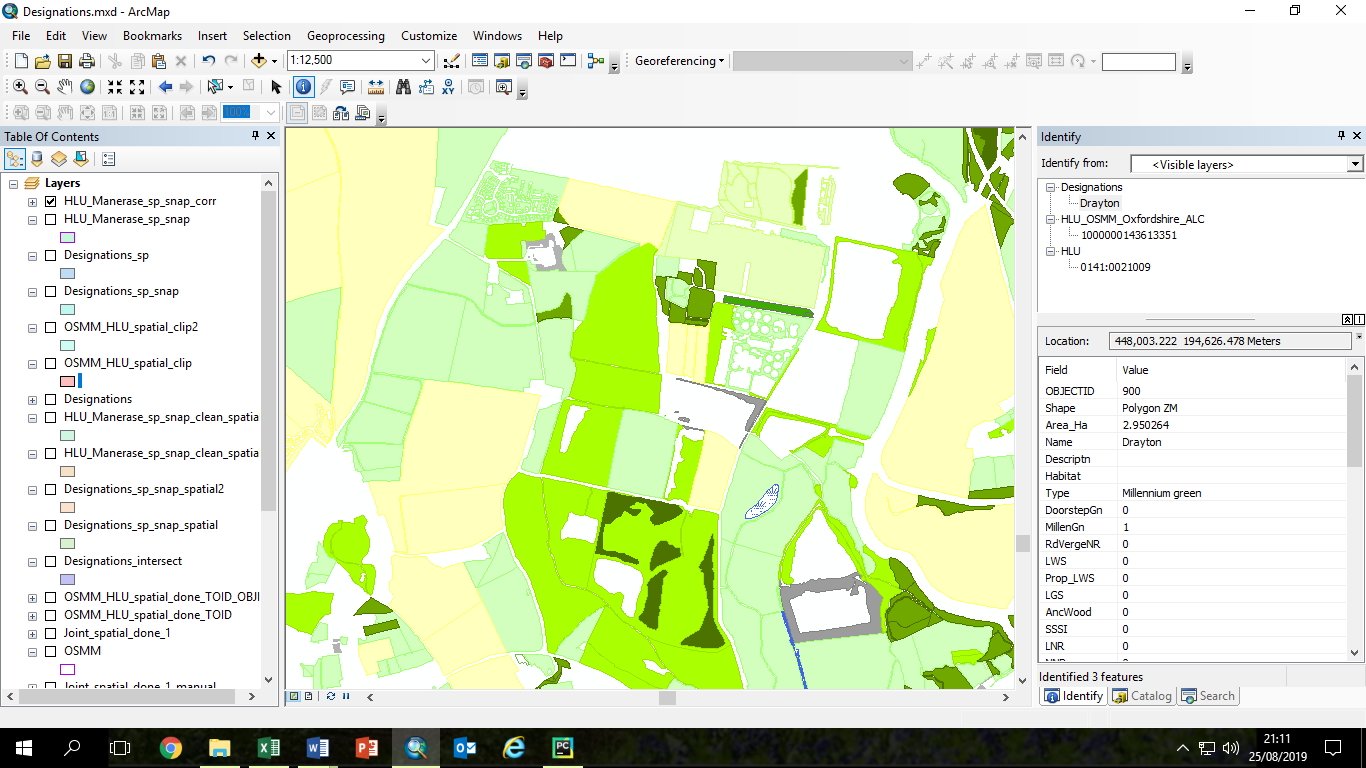
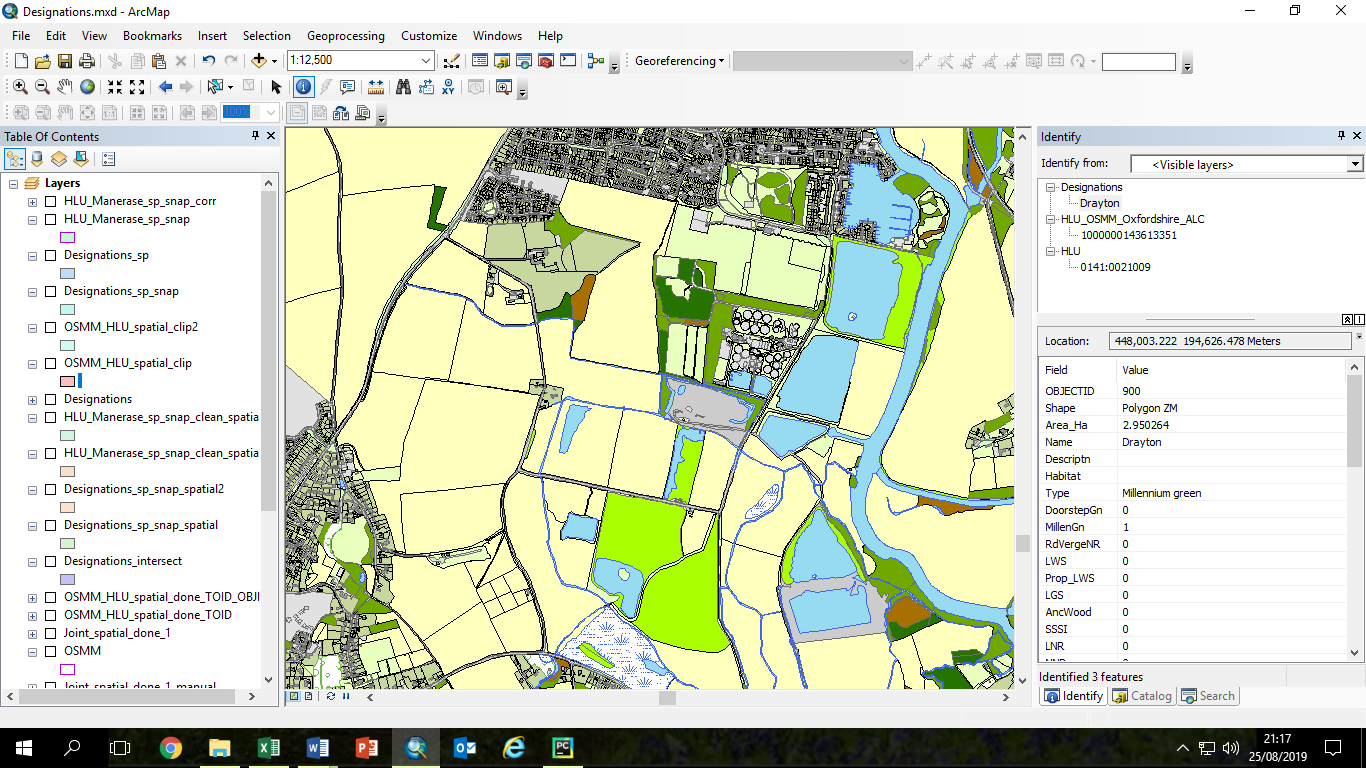
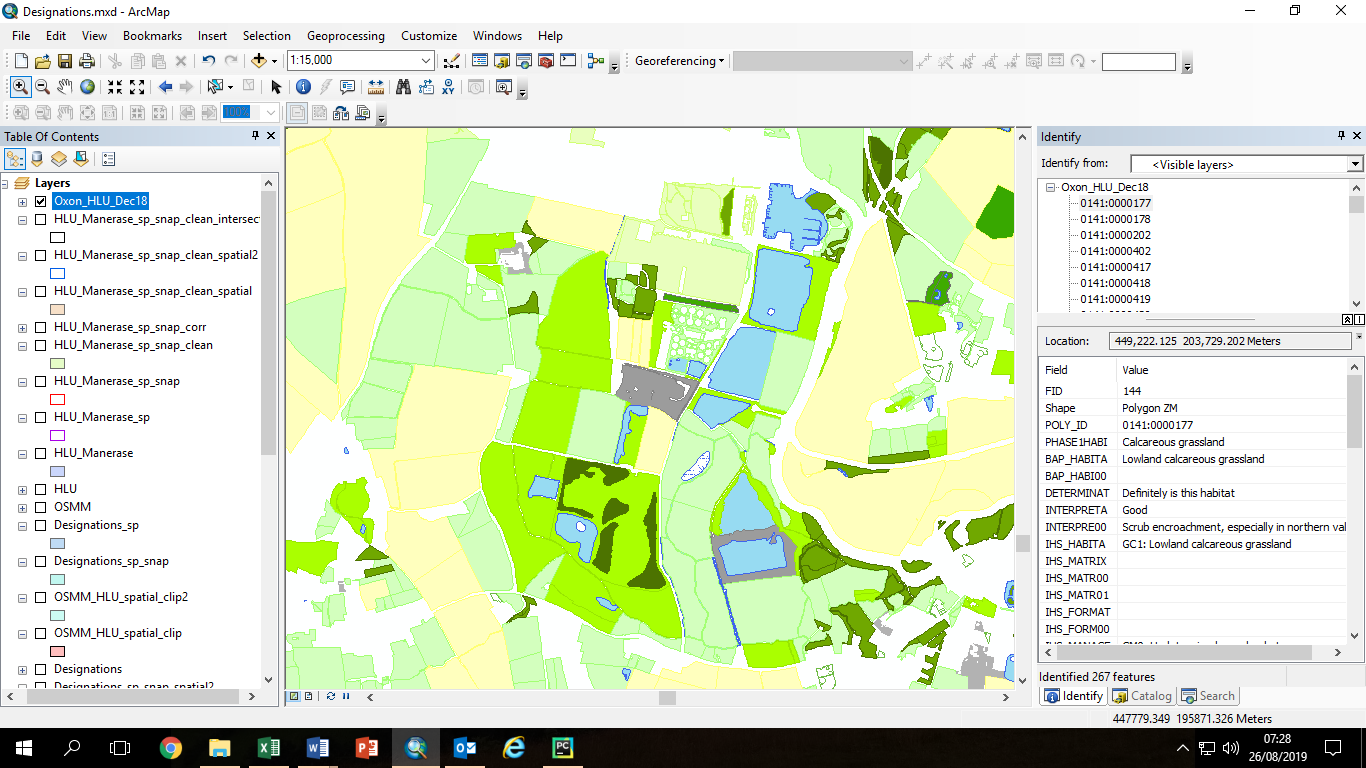
The method is based around a ‘tabulate intersection’ step to determine the percentage of new feature polygons within base map polygons.

## Prepare datasets

**Merge\_OSMM\_HLU\_ Preprocess.py**

**Delete OSMM landforms; delete HLU overlaps; erase manmade and water from HLU.**

1. Optional stage: clip to exact boundary of study area.
2. Optional stage: delete unnecessary attribute fields.
3. Delete landforms from OSMM -> OSMM\_noLandform NOTE: Pylons also overlap – originally these were not removed but I changed this as it was causing problems later in the code.
4. Delete HLU water features so we don’t get remnants left over later which obscure OSMM islands.
5. Erase OSMM manmade (buildings, roads, tracks) and water from HLU -> HLU\_Manerase Roadside is not erased, as this misses some semi-natural grassland and scrub etc.
6. Correct HLU overlaps with Union. **Note: this code may crash at the Union stage – if this happens, the union can be done manually in ArcGIS and the code can be re-started.**
7. HLU multipart to singlepart
8. Eliminate HLU overlap slivers (less than 1m2 – was originally also OR length/area >3 but I have removed this as I am not convinced it does not delete useful polygons.) This stage also deletes the small sliver gaps.
9. Delete remaining HLU stand-alone slivers
10. Delete remaining gaps (i.e. those that are genuine gaps, not slivers).
11. Delete identical shapes. If there are duplicate shapes with different habitats (there are a few of these in the Oxfordshire dataset) only one of these will be kept.



Examples from the South Abingdon / North Drayton area. Left: Original HLU layer omits urban areas and some water features. Middle: OSMM includes urban areas and woodland detail but not type of grassland. Right: pre-processed HLU (HLU\_Manerase) with manmade and water features erased, revealing shapes of lake and new housing estate.

## Main merge code – merge\_into\_base\_map\_V5a.py

First set up the input parameters in the first block of code. The steps in the main code are described below.

The code is designed to loop through a set of geodatabases, which we used for the different Local Authority Districts within the Oxford-Cambridge Arc. However it also works with only one geodatabase (which we used for Oxfordshire).

### Snap new features (e.g. HLU) to match base map features (e.g. OSMM) better

1. Only snap polygons that are not already identical (though only around 2000 out of 80000 HLU polygons are identical to OSMM after the pre-processing).
2. Snap new features to base map. Snap parameters are set up in the parameter input section. We use dmall snap distances (0.5 or 1m) to avoid distorting the edges of small linear features such as rides.
3. Correct overlaps created by the snapping process, by unioning and then deleting identical polygons.
4. Clean slivers from the snapped new features, by:
   1. Eliminating slivers less than 1m2
   2. Deleting stand-alone slivers <1m2

This is the most time-consuming part of the code –it can take 12 hours for a county the size of Oxfordshire. We experimented with densifying the two datasets before the snap, but this does not always work well and makes the snap operation even longer. Also it loses the original curves, replacing them with a series of short straight lines, and results in very large datasets (with far more points than previously).

### Decide which OSMM polygons should be split to include new HLU features

1. Save OBJECTID attribute and polygon areas to new attribute fields, for both the base map (Base\_ID) and the new features (New\_ID). These saved IDs will be used later, to transfer attributes across to the new merged shapes.
2. Use ArcGIS Tabulate intersection function to create a table with the percentage overlaps between the base map and the new features.
3. Create a new ‘Relationship’ attribute in the Tabluate Intersection table (“Base\_TI”) and use this to store the decision of whether or not to split polygons, based on a set of rules. Small overlaps are ignored, but larger overlaps mean that the base map (OSMM) polygon will be split (intersected with the new feature polygon outlines). However very large overlaps mean that the polygon will not be aplit, as the whole polygon will be assumed to match the new feature. The input spatial parameters ignore\_low and ignore\_high determine the threshold overlaps for ignoring or splitting polygons. There is also an overlap size threshold (significant\_size), above which polygons will be split even if the overlap is small.
   1. Very small overlap, less than ignore\_low (default 5%) AND overlap\_area < significant size (default 200 m2): ignore the new feature (habitat will remain same as base map). Relationship attribute set to ‘Base’.
   2. Overlap between ignore\_low and ignore\_high (default 5-95%): split the polygon so that the new feature outline is included within the base map polygon. Relationship attribute set to ‘Split’.
   3. Very large overlap, greater than ignore\_high (default 95%) – set the whole polygon to the same habitat as the new feature. Relationship attribute set to ‘New’.

**Examples: OSMM outlines in black and HLU outlines in red (selected HLU polygons in Cyan).**

|  |
| --- |
| **Example 1:** The large HLU arable polygon (Cyan outline) contains 4 medium-sized OSMM fields plus a very small OSMM field. The four medium OSMM fields and the small field all fit exactly into the HLU polygon so there is no need to split them. |
| **Example 2:** The HLU polygon (Cyan outline) isbetween 5% and 95% of the OSMM polygon, so the OSMM polygon will be split. |
| **Example 3. HLU (main area); OSMM (smaller area)**  The HLU woodland polygon (Cyan outline) is smaller than the OSMM scrub / woodland polygon but this is due to inaccurate mapping of the boundary. The overlap is less than the 95% threshold, so the whole polygon is assigned to the HLU habitat, and it is not split unless any of the smaller parts are greater than the ‘significant size’ threshold (default 200m2). |

|  |
| --- |
| **Example 4:**  **Top part**: a HLU ‘improved grassland’ field is split into a number of smaller OSMM areas including an area of scattered trees. These are all completely within the HLU field so they are not split. The small block of woodland and the small field next to it are assigned the OSMM classification later.  **Lower part**: an OSMM field is split into two HLU fields and includes a patch of OSMM trees in the north half which is not in HLU, and a HLU marsh in the lower half that is not included in OSMM.  The top half of the field is interpreted as the HLU neutral grassland. The small woodland will become OSMM woodland. The lower part of the field and the marsh are also assigned the HLU habitat types. |

### Add new HLU features to the OSMM base map (combine geometry)

We start from a clean copy of OSMM and modify it to include the new feature (HLU) polygons where they are justifiably different from OSMM – in other words the rows marked ‘Split’ in the tabulated intersection tables.

We have to be very careful here, because each base map (OSMM) polygon can have multiple corresponding rows in the Tabulate Intersection table, if the overlaps with the new features are complex. So some of the table joins are ‘one to many’ joins, which we have to be very careful with.

1. Make a copy of the Tabulate\_Intersection table, selecting only the rows for the polygons to be split.
2. For the polygons that we do not want to split (because they are over 95% overlapped by a new feature polygon), we need to make sure that the only the habitat attributes from the largest overlapping new feature are included (not any smaller ones round the edges). So make another copy of the Tabulate Intersection table for the un-split polygons, sort it by size, and delete those with a duplicate Base\_ID (this is what we set up earlier, by copying the original ObjectID). This keeps only the details for the main new habitat within that base map polygon.
3. Make a clean copy of the base map (OSMM) and add a ‘Relationship’ attribute field.
4. Join this to the table of split polygons, based on the Base ID, and set ‘Relationship’ to ‘Split’ for all those rows. This identifies all the base map (OSMM) polygons that need to be split.
5. Make a clean copy of the new features (HLU) and join it to the table of split polygons, using the new feature ID (New\_ID). This identifies all the new feature polygons that will form part of the new outlines in the merged file.
6. Select the new feature polygons marked ‘split’, and clip them using the outline of the base map polygons to be split. This will produce a set of new shapes to be merged into the base map, which we call ‘Joint\_spatial\_clip’.
7. Union these new shapes (which may be only partial, e.g. half of an OSMM field) into the base map polygons that they are splitting. This produces the new shapes for the split polygons, with both the base map attributes and (for the parts that are split), the new feature attributes.
8. Caution: This is the brain-melting bit. There are now two copies of the Relationship field - one from the unioned base map, with all rows 'split', and one from the clipped new features, with some polygons marked 'split' and some either null or blank. Some of the split polygons may have a large part that was tagged 'new' (because the overlap percentage was very high but there were also smaller parts of the same polygon that were tagged for splitting because they exceeded the significant size). So we now want to match the non-split parts of these polygons (joined in via the Union) with the correct ID from the TI table, so that attributes can be transferred later. We do this by sorting both the TI tables and the unioned clip file by size, and then joining the tables so that the split polygon parts are matched with the intersections of the same size, and copying the ID across.
9. Clean the clipped shapes, by eliminating and deleting slivers. This is the step where eliminating slivers may lead to small deviations from the integrity of the base map boundaries, including small polygons (< 1m2) in the original base map.
10. Another complicated bit. Create a tag ‘Not new’ to identify the parts of the split polygons that are base map only, i.e. not part of the new features. This is needed when transferring the new feature attributes back into the merged shapes, in step 13.
11. Erase the base map with the new joint shapes, and then merge the new shapes back into the base map. This has created a version of the base map that has polygons split where necessary to match the new features.
12. Convert to single part, check and repair geometry.

### Transfer attributes from OSMM and HLU to OSMM\_HLU layer

1. The merged dataset contains the base map attributes for all polygons, and the polygons that were split also contain the new feature attributes. However we now need to add the new feature attributes for the polygons that were not split but were marked as ‘new’, via a table join. This is a one to many join - each base polygon could be overlapped by more than one new feature polygon, but as these polygons were not marked for splitting, we want to use the ID of the largest intersection (i.e. ignoring slivers and insignificant areas) so we sort by size. The table join is done in two stages, because Relationship <> 'Not new' excludes NULLs.
2. The next few steps are a bit complicated but the end result is that all the attributes for the new features have been copied across to the appropriate polygons in the merged dataset.

## Interpreting data: OSMM\_HLU.py

This script assigns an overall habitat classification to each polygon as follows.

1. The OS mastermap attributes Descriptive Group, Descriptive Term and Make are used to assign new attribute field ‘OSMM\_hab’.
2. The Phase 1 habitat data (from the HLU habitat dataset) is used to assign new attribute field ‘HLU\_hab’.
3. The habitat based on a combination of OSMM and HLU is stored in ‘Interpreted Habitat’
4. The BAP classification (also from the HLU habitat dataset) modifies this where applicable, resulting in the final habitat interpretation in ‘BAP\_interpretation’.

It uses a fairly complex set of rules. It would need to be adapted if there are habitats present that we have not included, e.g. heather and bog.