**Office Title Normalization**

PKB: 20180514

See \Dropbox\CBDB\Ming official title Chn-Eng

for

Admin\_category\_&\_Admin\_unit\_codes\_draft\_pkb

This fills in the first three tables according to the Ma-Fuller proposal.

We will soon have a list of office titles to populate the empty tables in my draft, and more admin units, which we will add to the admin unit tables. (the fact that the numbering will be out of sequence does not matter).

NOTE 1: Our principle for coding  the names of offices and  administrative units, and to the degree possible categories of administrative units, is that the same name should have a single code number no matter what dynasty it belongs to. Some names (e.g. the Qing banners) will be unique to a dynasty.

NOTE 2: in the UCI Dictionary the category of local military officials includes subcategories by the names of provinces and other local units. Because we register the place as part of a posting, rather than as part of an office title (thus reducing the number of office titles greatly), we have not followed the UCI Dictionary in this regard. However, for offices in the capital bureaucracy that include place names (in other words do not denote postings to local government) the place name is kept as part of the administrative unit.

NOTE 3: We need a table for Alt\_names for office titles, not admin units

From MAF

Problem: Office titles in CBDB are unstructured strings.

Purpose: Code CBDB office title strings.

**Step 1: Defining an administrative units.**

A full administrative unit title should contain four parts: dynasty [D], classification(s) [C], administrative unit(s) [O], title [T]. For example: 清朝[D]文職[C]中央[C]吏部[O]文選清吏司[O]主事[T]. The structure of administrative unit title is: D+C+[O]+T. (I use the [O] here to indicate that the Administrative unit is going to be a multilevel object built from a succession of codes.)

Ma Ji’s proposal is a fine beginning, but we should refine it a bit

Probably we want to think of administrative unit titles as positions bound to a bureaucratic hierarchy defined by the D+C+O part.

The process of describing a bureaucratic entity with the name OA is to think of it as the limb, and climb back toward the trunk.

To what does OA report for this particular instance? That is, there may be various entities with the name OX but we know from the title of the administrative unit what the next level is.

So now we have OBOA. Climb back one more, and repeat until we have gotten to the dynasty: DOFOEODOC OBOA. Depending on the category of the unit directly attached to the dynasty (here OF), we can insert the category code: DCOFOEODOC OBOA.

**Step 2: Representing the Structure of administrative units.**

One way to represent the unit designated by the string DCOFOEODOC OBOA would be to just assign each such string a number. The table of office\_codes would then just use the one unique to define the administrative unit of which the office is a part. This has the virtue of simplicity, but it does not handle missing data at all gracefully. We would need to generate a huge set of codes to describe the incomplete data we find in texts. (See more about this below.)

Instead of a single code, we could use the composite string DCOFOEODOC OBOA to represent the unit with the name OA. However, my sense is that it will be easier to debug and be better normalized if we take the name codes out of the string that defines the unit.

Consider OC: it is a name bound to a tree structure (DOFOEODOC), but that name may also be bound to different structures at different times. Still, this is one unique instance of a bureaucratic unit.

I believe that the best way to represent this unique instance is what we already are using for our administrative unit tree structure and replace the administrative unit name codes with numbers. I don’t think this gets us in any trouble, although I am open arguments to the contrary.

For the Ming, if there are 7 categories hung off of the Dynasty, then those categories get designations 19,1-7. If category 4 has 13 subcategories, and there are 6 bureaucratic units in subcategory 8, then the bureaucratic unit with the name code OA (from the Admin\_name\_codes table) and listed in position 5 as part of unit 6 among the 6 units in subcategory 8 will have a record in the Admin\_Codes table:

OA 19  **4 8**  6 5 Begin-year -End-year Notes  
 OA (the name) will have a list of such records

Note that the denormalized tree can be reconstructed, since we will have a record  
 OB 19 **4 8**  6 Begin-year -End-year Notes

where the begin and end years match

Part of the point of using numbers rather than the office name codes is to help keep in mind that the fundamental point that the name is an attribute of an entity defined crucially by its position in the hierarchical structure, even though the name hopefully does suggest the function of the unit. The fact that the name appears within various hierarchical structures suggests that its function does adjust to context. Still, if people want to keep the superior administrative unit name codes as parts of the record (at least while we are trying it out), that is fine.

Revising the bureaucratic structure will be messy under any circumstance. If the unit named OB should not have been placed where it was, then all the administrative subunits under it (included the unit with the name OA) will need to redefined, as will all the postings to offices in those units. The structure of the record for an administrative unit will neither complicate nor simplify fixing such mistakes (but, again, I can be persuaded otherwise if I am missing something here).

**Step 3: Structure of Offices.**

An office with title code T (from the Office\_name\_codes table) that is attached to unit with the name OA that has the entry in the Admin\_unit\_codes table [OA 19  **4 8**  6 5] would have a record in the Office\_codes table:

T OA 19 4 8 6 5 Begin-year -End-year Notes   
 [OA 19 4 8 6 5] uniquely defines the particular, relevant instance of OA

T (the name) also will have a list of such records for where the name appears in different   
 administrative contexts that defines its historical arc.

**Step 4: Structure of Postings.**

Finally, for a posting, we will have T and perhaps OA and all the way up, but we may be lacking data. What I suggest may work is that for each T we find out the longest string of codes that uniquely defines the office (or T OA or T OA OB), fill in the information we have at the appropriate level, and then put 0 in the missing higher level values unless we know the dynasty or that T always goes under a particular category and subcategory:

The POSTED\_TO\_OFFICE record would look like:

PostingID PersonID T OA 19  **0 0** 0 0 Begin-year -End-year Notes  
 Here we need the OA name code because we may not be able to uniquely define the administrative unit with the name OA to which the office belongs.

The POSTED\_TO\_ADDR record would look like:

PostingID PersonID T OA 19  **0 0** 0 0 OfficeADDRID Begin-year -End-year Notes

If there is only one record for the name OA that begins with [T 19] in the Admin\_unit\_codes table, then of course we can fill in the additional data (but mark it as filled in from the table)

**The Tables**

1. Admin\_category\_codes (Ma Ji’s C\_OT\_CLS table, modified to match the Ming office titles)

1. c\_admin\_cat\_id
2. c\_admin\_cat\_hz: Chinese name.
3. c\_admin\_cat\_trans: English name.
4. c\_admin\_subcat\_id: The Ming example seems to have a two-part structure as in

|  |  |
| --- | --- |
| 中央輔佐官署類 Central Administration Assistance | 寺監門 Courts and Directorates |

1. c\_admin\_subcat\_hz: office title, classification Chinese name.
2. c\_admin\_subcat\_trans: office title, classification English name.
3. c\_admin\_subcat\_start\_year: (Probably need to add c\_ admin\_subcat\_start\_nianhao)
4. c\_admin\_subcat\_end\_year: (Probably need to add c\_ admin\_subcat\_end\_nianhao)
5. c\_admin\_cat\_notes
6. c\_admin\_cat\_source\_id

2. Admin\_unit\_name\_codes (Ma Ji’s C\_OT\_OFC table)

1. c\_admin\_unit\_name\_id
2. c\_admin\_unit\_name\_hz
3. c\_admin\_unit\_name\_py
4. c\_admin\_unit\_name\_trans
5. c\_admin\_unit\_name\_desc\_hz
6. c\_admin\_unit\_name\_desc\_trans
7. c\_admin\_unit\_name\_start: (Probably need to add c\_ admin\_unit\_name\_start\_nianhao, dy?)
8. c\_admin\_unit\_name\_end: (Probably need to add c\_ admin\_unit\_name\_end\_nianhao, dy?)
9. c\_admin\_unit\_name\_notes
10. c\_admin\_unit\_name\_source\_id

3. Admin\_unit\_codes

1. c\_admin\_unit\_name\_id
2. c\_admin\_unit\_dy
3. c\_admin\_cat\_id
4. c\_admin\_subcat\_id
5. c\_admin\_superior\_unit1\_id (this is the topmost unit in the tree, attached to dynasty)
6. c\_admin\_superior\_unit1\_name\_id (if people want it)
7. c\_admin\_superior\_unit2\_id
8. c\_admin\_superior\_unit2\_name\_id (if people want it)
9. c\_admin\_superior\_unit3\_id
10. c\_admin\_superior\_unit3\_name\_id (if people want it)
11. c\_admin\_superior\_unit4\_id
12. c\_admin\_superior\_unit4\_name\_id (if people want it)
13. c\_admin\_superior\_unit5\_id
14. c\_admin\_superior\_unit5\_name\_id (if people want it)
15. c\_admin\_superior\_unit6\_id
16. c\_admin\_superior\_unit6\_name\_id (if people want it)
17. c\_admin\_unit\_start\_year
18. c\_admin\_unit\_end\_year
19. c\_admin\_unit\_source\_id
20. c\_admin\_unit\_notes: (this can include the original title from which one is working to create the parsed, segmented version)  
      
    In our current OFFICE\_TYPE\_TREE, I use a text string of numbers for all of the superior unit codes. Some operations are easier using the string approach, which also is more flexible in that it can be arbitrarily long. The only issue is that one has to decide whether one needs just 99 branches at some level rather 999. The lazy person’s approach would be to include the string as well.

4. Admin\_unit\_altname\_codes  
PKB – we have altnames for office titles in the UCI dictionary, not for Admin units.  
This table will look like either the Admin\_unit\_name\_codes table or the Admin\_unit\_codes table with one additional name id added, depending on whether the alternative names apply to all instances of the name or only to some particular instances. I suspect the latter may be the case but do not know for sure.

5. Office\_name\_codes (Ma Ji’s C\_OT\_TIT:

1. c\_office\_name\_id
2. c\_office\_name\_hz
3. c\_office\_name\_py
4. c\_office\_name\_trans
5. c\_office\_name\_start\_year (nianhao? dy?)
6. c\_office\_name\_end\_year (nianhao? dy?)
7. c\_office\_name\_notes
8. c\_office\_name\_source\_id

6. Office\_codes

1. c\_office\_name\_id
2. c\_admin\_unit\_name\_id
3. c\_admin\_unit\_dy
4. c\_admin\_superior\_unit1\_id (this is the topmost unit in the tree, attached to dynasty)
5. c\_admin\_superior\_unit2\_id
6. c\_admin\_superior\_unit3\_id
7. c\_admin\_superior\_unit4\_id
8. c\_admin\_superior\_unit5\_id
9. c\_admin\_superior\_unit6\_id
10. c\_office\_start\_year
11. c\_office\_end\_year
12. c\_office\_source\_ id
13. c\_office\_notes: (this can include the original title from which one is working to create the parsed, segmented version)

7. Posting\_data

1. c\_person\_id
2. c\_posting\_id

8. Posted\_to\_office\_data (one posting may involve more than one office)

1. c\_posting\_id
2. c\_office\_name\_id
3. c\_admin\_unit\_name\_id (it is possible that c-j could all be set to 0)
4. c\_admin\_unit\_dy
5. c\_admin\_superior\_unit1\_id
6. c\_admin\_superior\_unit2\_id
7. c\_admin\_superior\_unit3\_id
8. c\_admin\_superior\_unit4\_id
9. c\_admin\_superior\_unit5\_id
10. c\_admin\_superior\_unit6\_id
11. c\_posted\_office\_sequence (for when we know sequence but not actual years)
12. c\_posted\_office\_start\_year
13. c\_posted\_office\_end\_year
14. c\_posted\_office\_source\_id
15. c\_posted\_office\_notes  
    etc. (posting type, whether he made it to office, and the other information captured in our current tables)

9. Posted\_to\_office\_addr\_data (one office may involve more than one address)

1. c\_posting\_id
2. c\_office\_name\_id
3. c\_admin\_unit\_name\_id (it is possible that c-j could all be set to 0)
4. c\_admin\_unit\_dy
5. c\_admin\_superior\_unit1\_id
6. c\_admin\_superior\_unit2\_id
7. c\_admin\_superior\_unit3\_id
8. c\_admin\_superior\_unit4\_id
9. c\_admin\_superior\_unit5\_id
10. c\_admin\_superior\_unit6\_id
11. c\_posted\_office\_sequence (for when we know sequence but not actual years)
12. c\_posted\_office\_addr\_id (or, in the new version c\_addr\_name\_id + c\_addr\_loc\_id + c\_addr\_instance\_id)
13. c\_posted\_office\_addr\_source\_id
14. c\_posted\_office\_addr\_notes