**Instructions for Providing Pre-Analyzed Hex Data for Use on the NatureServe Explorer Website**

*Document Version 3.1 | Drafted By: Jason McNees, Dave Hauver | Updated by: Cameron Scott | Version Date: 07/29/2022*

In order to load spatial data into Explorer, programs need to provide a CSV file or an Esri File Geodatabase table that records the intersections between their species presence records and the one square mile hexagons in the Nested Hexagon Framework. If programs need to obfuscate their data to either 7, 49 or 343 square miles, they can analyze their data against the NHF cogs or wheels instead of the one square mile hexagons. The NHF wheels are used for both 49 and 343 square miles.

Data should only be provided for plants, animals, and fungi. Programs should not provide ecological data at this time.

**Data Format**

The final product should be an Esri File geodatabase which includes the columns from the SQL further down in the document. If data is being provided obfuscated to NSX Pro hexagons then delivery can be in CSV format instead of a geodatabase and should include **HEXAGON\_ID** (The NHF hexagon ID. For example, 120W-40N-0001-01-2).

If data is being obfuscated to 7 square miles, provide a COG\_ID instead of HEXAGON\_ID:

COG\_ID = 120W-40N-0001-01

If data is being obfuscated to 49 square miles, provide a WHEEL\_ID instead of HEXAGON\_ID:

WHEEL\_ID = 120W-40N-0001

If data is being obfuscated to 343 square miles, provide a SUMMARY\_HEX\_L1\_ID instead of HEXAGON\_ID. The SUMMARY\_HEX\_L1\_ID is found in the wheels layer. Duplicate results do not need to be removed prior to delivery.

120W-45N-1580-L1

If data is being provided for a mix of scales (1 sq mi, 7 sq mi, 49 sq mi, and/or 343 sq mi), a separate CSV file can be provided with the results for each one (see note at the end of the "Analyzing Data" section below). However, these should be mutually exclusive so that EOs that are being obfuscated to a larger scale do not appear in tables for any other scale (i.e. EOs being intersected to the 7 mile grid should not also appear in results for the 1 mile intersection, or in the results for EOs being intersected with the 49 mile grid).

If no precise data is being provided, only obfuscated data, CSV format is preferred.

**Obfuscating Data**

For data being obfuscated, this can be done in many ways. If using ArcGIS Desktop or Pro, the Spatial Join tool (<https://pro.arcgis.com/en/pro-app/tool-reference/analysis/spatial-join.htm>) can be used to analyze the species presence features against the NHF features. The resulting feature class can be converted to a table which contains the desired columns. (Note: as mentioned in the "Data Format" section above, if EO data is being obfuscated to a mix of hex scales, a separate EO layer should be created with the records that will be intersected with each hex size that are mutually exclusive – the same EO should not be intersected against more than one grid size).

The Nested Hexagon Framework can be downloaded from <https://github.com/NatureServe/nested-hexagon-framework>.

Note that the Nested Hexagon Framework geodatabase contains two sets of features. One set of features are the features that exist in the original Nested Hexagon Framework developed by the University of Kansas:

* nhf\_hexes
* nhf\_cogs
* nhf\_wheels

NatureServe extended the framework to cover Hawaii, Northern Canada, the Caribbean, Bermuda, and Central America. These are stored in separate layers:

* nhf\_hexes\_na
* nhf\_cogs\_na
* nhf\_wheel\_na

If the spatial data being analyzed is located in these areas, the analysis should be performed twice, once against the original features and once against the features within the extended coverage area. It is okay to provide the results in two separate files, or if desired, they can be merged into a single file.

In ArcGIS, tables can be converted to CSV format using the Table to Table tool (<https://pro.arcgis.com/en/pro-app/tool-reference/conversion/table-to-table.htm>). To do this, set the “Output Location” to be a folder (instead of a geodatabase), and in the “Output Table” parameter, add a “.csv” extension to the name of the output table. However, we can also accept a table within a file geodatabase if programs are unable to convert the table to a CSV. Please use the following naming convention for the output CSV or geodatabase table: [Subnation]\_Explorer\_Hex\_Dataset\_[MMDDYYYY].csv

Note: if multiple tables with different scales of results are being provided, the naming convention of the tables above can be modified to be "\_1miHex\_", "\_7miHex\_" or "\_49miHex\_" to distinguish them.

**Species Presence Data Sets**

**EOs stored in Biotics**

If analyzing EOs which are stored in Biotics, export the plant and animal EOs from Biotics. Biotics Query Builder can be used to export data to a file geodatabase. The following query can be used as-is or modified as necessary if additional restrictions are needed. If providing precise data this export can be used for delivery.

NOTE: This query excludes records marked "Do Not Exchange" in the last line. If there are no issues providing these records at the hexagon scale, and it is preferred they be included, that criteria should be removed.

*SELECT*

 *eo\_shape.shape,*

 *('EO.' || eo.eo\_ou\_uid || '.' || eo.eo\_seq\_uid) as RECORD\_UID,*

 *('ELEMENT\_GLOBAL.' || egt.element\_global\_ou\_uid || '.' || egt.element\_global\_seq\_uid) as*

 *ELEMENT\_GLOBAL\_UID,*

 *subnation.subnation\_code subnation,*

 *egt.g\_rank,*

 *egt.rounded\_g\_rank rnd\_g\_rank,*

 *est.s\_rank,*

 *est.rounded\_s\_rank rnd\_s\_rank,*

 *estdsens.data\_sensitive\_cd s\_datasen,*

 *case*

 *when eodsens.data\_sensitive\_cd = 'Y' then 'Y'*

 *when eosub\_dsens like '%Y%' then 'Y'*

 *when eodsens.data\_sensitive\_cd = 'C' then 'C'*

 *when eosub\_dsens like '%C%' then 'C'*

 *else 'N'*

 *end eo\_datasen,*

 *eodsens.data\_sensitive\_cd eo\_princ\_datasen,*

 *eo.eosub\_dsens,*

 *estdsenscat.data\_sensitive\_category\_desc s\_datasen\_cat,*

 *delimlist('select data\_sensitive\_category\_desc*

 *from (select eodsens.data\_sensitive\_category\_desc*

 *from eo eosb*

 *join d\_data\_sensitive\_category eodsens on eosb.d\_data\_sensitive\_category\_id = eodsens.d\_data\_sensitive\_category\_id*

 *where eosb.principal\_eo\_shape\_id = ' || eo.shape\_id || '*

 *union select ''' || eodsenscat.data\_sensitive\_category\_desc || ''' from dual*

 *) tmp group by data\_sensitive\_category\_desc',', ') eo\_datasen\_cat,*

 *eodsenscat.data\_sensitive\_category\_desc eo\_princ\_datasen\_cat,*

 *delimlist('select eodsens.data\_sensitive\_category\_desc*

 *from eo eosb*

 *join d\_data\_sensitive\_category eodsens on eosb.d\_data\_sensitive\_category\_id = eodsens.d\_data\_sensitive\_category\_id*

 *where eosb.principal\_eo\_shape\_id = ' || eo.shape\_id || '*

 *group by eodsens.data\_sensitive\_category\_desc',', ') eosub\_dsens\_cat,*

 *eorank.basic\_eo\_rank\_cd eorank\_cd,*

 *eo.last\_obs\_date lastobs\_d*

*FROM*

 *(select eo.\*,*

 *delimlist('select eodsens.data\_sensitive\_cd*

 *from eo eosb*

 *join d\_data\_sensitive eodsens on eosb.d\_data\_sensitive\_id = eodsens.d\_data\_sensitive\_id*

 *where eosb.principal\_eo\_shape\_id = ' || eo.shape\_id,',') as eosub\_dsens*

 *from eo) eo*

 *join eo\_shape on eo.eo\_id=eo\_shape.eo\_id*

 *join element\_subnational est on eo.element\_subnational\_id=est.element\_subnational\_id*

 *join element\_national ent on est.element\_national\_id=ent.element\_national\_id*

 *join element\_global egt on ent.element\_global\_id=egt.element\_global\_id*

 *join scientific\_name sn on egt.gname\_id=sn.scientific\_name\_id*

 *join subnation on est.subnation\_id = subnation.subnation\_id*

 *join d\_name\_category nc on sn.d\_name\_category\_id=nc.d\_name\_category\_id*

 *left outer join d\_basic\_eo\_rank eorank on eo.d\_basic\_eo\_rank\_id = eorank.d\_basic\_eo\_rank\_id*

 *left outer join d\_id\_confirmed id on eo.d\_id\_confirmed\_id = id.d\_id\_confirmed\_id*

 *left outer join d\_data\_sensitive estdsens on est.d\_data\_sensitive\_id = estdsens.d\_data\_sensitive\_id*

 *left outer join d\_data\_sensitive eodsens on eo.d\_data\_sensitive\_id = eodsens.d\_data\_sensitive\_id*

 *left outer join d\_data\_sensitive\_category estdsenscat on est.d\_data\_sensitive\_category\_id = estdsenscat.d\_data\_sensitive\_category\_id*

 *left outer join d\_data\_sensitive\_category eodsenscat on eo.d\_data\_sensitive\_category\_id = eodsenscat.d\_data\_sensitive\_category\_id*

*WHERE*

 *nc.d\_name\_category\_id in (1, 2, 4, 5, 6, 7, 19, 24)*

 *and (id.id\_confirmed\_cd <> 'N' or id.id\_confirmed\_cd is null)*

 *and (eo.do\_not\_exchange\_ind <> 'Y' or eo.do\_not\_exchange\_ind is null)*

*;*

**Data from other sources**

If data originates from a data source other than Biotics, the RECORD\_UID value can be formatted as necessary. However, the taxonomy must be resolved against Biotics, and an ELEMENT\_GLOBAL\_UID value must be included in the data (as outlined in the query above).

**Delivery of Data to NatureServe**

Once the analysis is complete, please add the CSV or geodatabase table(s) created above into a password-protected Zip file that follows the naming convention: [Subnation]\_Explorer\_Hex\_Dataset\_[hex/precision scale]\_[MMDDYYYY].zip.

Then, upload that Zip file to NatureServe's file transfer site using the instructions in the attached document. Note for ERT programs, the login credentials for the ERT and guest folders are different. Once the file is there, please notify NatureServe by e-mailing ExplorerSupport@natureserve.org with the following subject line, and be sure to include the password for the Zip file in the body of the message: [Subnation] Explorer Hex Refresh Dataset is Posted – [MM/DD/YYYY]

If you are unable to use our transfer site and prefer to post it to a different site, please e-mail ExplorerSupport@natureserve.org with the same subject line, along with a link to the file location, and the Zip file password. That e-mail address can also be used if you have any trouble posting the file to the NatureServe transfer site and need assistance.