Grazinglands Resource Analysis System (GRAS)
Model and Data Services
Specification

Input Data
Computational Logic
Output Data

January 6, 2016

Department of Civil and Environmental Engineering
Colorado State University
Fort Collins, Colorado

Version 20160106

Table of Contents

Introduction: Grazing Management (GRAS) Model and Data Services

This document contains the process and data definitions for the computational and data services of the USDA Natural Resource Conservation Service (NRCS) Grazing Management module called the Grassland Resource Analysis System (GRAS). GRAS integrates the concepts and logic of farm/ranch grazing unit level conservation planning and the analysis of livestock-related resource concerns, specifically the concern Livestock Production Limitation: Inadequate Feed and Forage. GRAS inventories forage production for a set of grazing units, inventories animal herd forage demand, schedules animal herds on grazing units, and computes an animal forage balance for the grazing system. GRAS also can compute a quick stocking rate and support on-site data collection for forage inventory. The GRAS model and data services support this workflow. Output from these services also feeds tailoring the NRCS Prescribed Grazing conservation practice for the grazing units in the system.

A NRCS conservation plan contains conservation practices scheduled on planning land units (PLUs). A grazing unit is an area that is grazed on a particular schedule and usually is bounded by a fence to contain the grazing animals. A grazing unit is synonymous with a PLU.

A grazing system contains one or more PLUs (grazing units). A rancher rotates grazing animals (one or more herds) through PLUs according to a grazing schedule (one or more years). The Prescribed Grazing Practice bounds and describes the grazing system.

Instead of using the PLU entity for associating animal and forage data, GRAS will use the area of analysis (AoA) entity. The primary reason is to enable analysis of different scenarios (benchmark, alternative, planned) and different grazing unit configurations (geometries). In most cases, an AoA is congruent with a PLU.

The NRCS planner inventories forage production on each grazing unit (AoA) and forage demand of the animals to be included in the grazing system. Forage production and demand is computed on a daily basis, which can be aggregated to weekly, monthly, and annual. GRAS computes an animal-forage balance based on forage production and the demand represented in the grazing schedule. Imbalances can be mitigated by supplemental feed or adjusting the stocking rate.

This document is organized by services and components. Services are web services intended to interact with a requesting application, for example, with the NRCS Customer Service Toolkit. The application sends a request payload (expected by the service), and the service returns a results payload to the application. Also sometimes/often a service may interact with other services.

Services contain one or more components. Components encode one or more particular processes (algorithms, computational logic, data access, etc.). Computational efficiency drives component design, whereas requirements of expected requesting applications drive service

scope and design. Therefore we should consider this document to be at least somewhat fluid in the early phases of WQM development as business and system requirements settle.

The NRCS has developed an extensive library of GRAS business requirements, rules, and workflows, which provide the primary reference for developing these service specifications.

Overview of the GRAS Workflow

Preliminary Steps – User runs an application to open a customer folder containing geospatial land units (planning land units or PLUs and decides to create a grazing system spanning several of the PLUs. Employing the PLU layer as a template, the user creates a layer of polygons designated as areas of analysis (AoAs). This is the starting point for GRAS.

Step 1 – The user intersects AoA polygons with Soil Survey Geogaphic Database (SSURGO) mapnuit polygons. Each polygon created by the intersection becomes a forage inventory site (FIS). When this occurs, the application calls a GRAS service to get data for attributing these sites. The application uses the GRAS payload to populate dialogs as the user goes through the process to attribute each FIS. The user may split or merge FIS polygons. The user may attribute to the FIS level (basic forage inventory) or to the FIS component level (detailed forage inventory), but the application has enough data from GRAS to enable the user to do both. This step results in one or more FIS for each AoA, estimated production for each FIS/component, and associated plant growth curves. The resulting payload is saved to the application database.

Step 1a – The user returns to the FIS step to complete or edit existing data. The application saved data from its database and calls the GRAS service to get data that may be used for changes or fill in gaps.

Step 2 (optional) – The user at this point may want to have a quick stocking rate (QSR) calculated for the grazing system, including a breakdown by AoA. When the user has recorded measured production for the FISs in the AoAs of the grazing system, the application calls a GRAS service to calculate and return QSRs for the affected AoAs including a grazing system total.

Step 3 – To proceed towards calculating forage animal balance, the user sets a forage partition profile (FPP) for each AoA each year in the grazing system. A GRAS service returns a list of FPPs applicable to the location of the grazing system. A FPP designates periods when an AoA is available for grazing during the year. Other periods are not available for grazing and have two types: restricted use (no grazing, forage often accumulates for later grazing, but may be reduced by burning or other treatment) and harvest roughage (no grazing, forage accumulates and is harvested at the end of the period for hay, silage, or other type of roughage)

Step 4 – Next the user identifies and creates areas within any of the AoAs in the grazing system to reflect adjustments to available forage. For example, an area infested by weeds may reduce available forage to some percentage of normal. These are called forage adjustment areas (FAAs), which are factored into calculating forage available for grazing.

Step 5 – The user defines animal groups comprising the herds that will graze available forage. GRAS services provides the application with animal data for defining these herds and calculating their demand for forage.

Step 6 – Any of the herds created may involve breeding and the birth of offspring, and the user may add breeding animals and offspring to the affected herds with herd forage demand adjusted accordingly. A GRAS service provides computational support to this step.

Step 7 – The user often will establish a grazing schedule by assigning one or more herds to the AoAs in the grazing system during periods when AoAs are available for grazing. A GRAS service calculates a period forage animal balance to assist matching animal demand with forage supply.

Step 8 – With forage supply and animal demand defined for the grazing system, the application calls GRAS services to calculate the forage animal balance. Two calculation options: one without grazing schedule or one when the user has created a grazing schedule.

Service GRAS-1a: Create Forage Inventory Sites, Get Ecological Sites and Estimated Production for an Area of Analysis (FISProdESD)

Purpose: Create forage inventory sites (FISs) for an area of analysis (AoA) and get estimated production from associated Ecological Site Information System (ESIS) ecological site plant communities or associated SSURGO soil components or mapunits.

To calculate forage supply, the GRAS application user must delineate one or more forage inventory sites within the AoAs of a grazing system. The user can choose one of six methods:

- Intersect the AoAs in the grazing system with SSURGO soil mapunit layer; each intersected polygon becomes a FIS within an AoA corresponding to a soil mapunit.
- User splits or merges polygons created by AoA by soil mapunit intersection above.
- Copy an AoA boundary to become a FIS
- Intersect AoAs with polygons of other layers (e.g. a state ecological site layer); each intersected polygon becomes a FIS within an AoA.
- Upload GPS coordinates and intersect with AoAs; each intersecte polygon becomes a FIS within an AoA.
- User-digitized polygons intersected with AoAs, each intersected polygon becomes a FIS within an AoA.

These six methods can be abstracted into three: (1) AoA boundary is the FIS polygon, (2) AoA contains one or more FIS polygons created by AoA x soil mapunit intersection, and (3) AoA contains one or more FIS polygons created by AoA x soil mapunit x user geometry intersection.

Any of the methods above requires the creation of a table containing soil components relevant to the FIS. Many soil components link to ecological sites (ESDs) in ESIS containing estimated dry matter production for their associated plant communities. SSURGO components themselves also can contain forage dry matter production values, which serve as a backup if no production values exist in ESIS.

This service consumes an application request payload to create one or more FIS per AoA, get associated ecological sites, and get estimated dry matter production for associated plant communities and soil components. The results payload returns data enabling the requesting application to populate a choice list for associating estimated dry matter production to a FIS or FIS soil component.

This service applies only to grazing units (AoAs) having natural plant communities (non-cultivated) and land use range and these land uses with grazing modifiers: Forest, Protected, Other Rural Land, and Associated Agricultural Land.

Service Description:

Create forage inventory sites (FISs) and get estimated forage production for the plant communities and their ecological states from ecological site descriptions (ESDs) .

Service Signature

Request Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
aoa_geometry … geospatial coordinates, one set per AoA, Area of Analysis Polygon Geometry
fis_method … integer, Method for Creating Forage Inventory Site; choices are 1 - FIS is AoA boundary, 2 – FIS is AoA x mapunit intersection boundary, 3 – FIS is AoA x user supplied geometry intersection boundary
aoa_land_use … integer, corresponding to NRCS land_use_id; choices for this service are 2 – forest, 3 – range, 5 – Protected, 9 – Other Rural Land, and 10 – Associated Agricultural Land
est_prod_method … integer, value is 1 for this service; Method to Get Estimated Forage Production; for this service: 1 – from ESIS ecological site; see GRAS-1b for method 2 – from ESIS forage suitability group, and GRAS-1c for method 3 – from SSURGO
user_fis_geometry … geospatial coordinates, User Supplied Geometry for Creating Forage Inventory Sites; can be one or more lines or polygons
planner_id … character varying(23), Application User Identifier
inventory_date … Date (yyyy-mm-dd), Forage Inventory Date

Result Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
planner_id … character varying(23), Application User Identifier
inventory_date … Date (yyyy-mm-dd), Forage Inventory Date
fis_type … integer, Forage Inventory Site Type; values are 1 – Single or 2 - Multiple
fis_id … integer, one or more per AoA depending on FIS method, Forage Inventory Site Identifier
fis_geometry … geospatial coordinates, one or more sets for the AoA, Forage Inventory Site Geometry
es_id … character varying(60), one or more per FIS, Ecological Site Identifier
es_range_name … character varying(120), Ecological Site Name, name applied if ecological site identifier begins with R
es_forest_name … character varying(120), Ecological Site Name, name applied if ecological site identifier begins with F
es_pctfis … double precision, Percent of Forage Inventory Site
es_rsprod … bigint, Estimated Forage Production
plant_community_id … numeric(2,0), one or more plant communities per ecological site, Plant Community Identifier
plant_community_name … character varying(100), Plant Community Name
plant_communty_sequence … numeric(2,0), Sequence of the Plant Community and Its State in the Ecological Site
state_id … numeric(2,0), Plant Community Ecological State Identifier
state_name … character varying(100), Name of the Ecological State of the Plant Community
es_est_prod … numeric(5,0), Representative Forage Production for

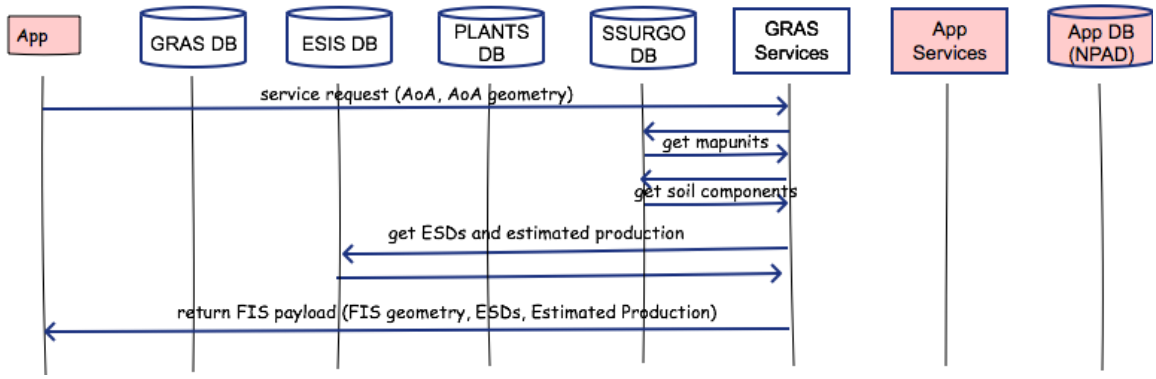
the Plant Community

Reference Data Sources

SSURGO Data Mart
- mapunit table
- component table
- coecoclass table

ESIS Data Mart
- Ecological_Sites table
- Plant_Communities table
- Range_Annual_Production table

GRAS-1a: Create Forage Inventory Sites (FIS), Get ESDs and Estimated Production



Component

1. **Create Forage Inventory Sites and Get ESD-Based Estimated Production (FISESDProd)**

1.1. Inputs

AoAId	1	2	3
aoa_geometry	[40.6735513978082, -103.809573763755], [40.6607911607823, -103.82648240938], [40.6735513978082, -103.82648240938], [40.6607911607823, -103.809573763755]	[40.6689809721788, -103.8472001256], [40.6434563076883, -103.896295279396], [40.6689809721788, -103.896295279396], [40.6434563076883, -103.8472001256]	[40.6989209302277, -103.972512930964], [40.6502291091709, -104.086496085249], [40.6989209302277, -104.086496085249], [40.6502291091709, -103.972512930964]
aoa_land_use	3	9	4
est_prod_method	1	1	1
fis_method	1	2	3
user_fis_geometry			[40.710307599928, -104.085122794275], [40.6295829439063, -104.002038687843]
planner_id	Dana Smith	Dana Smith	Jan Jones
inventory_date	20141003	20140915	20150223

1.2. Data
- SSURGO
  - mapunit
    - mukey
  - component
    - mukey
    - cokey
    - otherph
    - localphase
    - majcompflag
    - comppct_r
  - coecoclass
    - cokey
    - ecoclassid
    - ecoclassname
- ESIS
  - Ecological_Sites table
    - es_type
    - es_mlra
    - es_mlru
    - es_site_number
    - es_state
    - forest_sitetree1_vernacular
    - forest_sitetree2_vernacular
    - forest_siteshrub1_vernacular
    - forest_siteshrub2_vernacular
    - forest_siteherb1_vernacular
    - forest_siteherb2_vernacular
    - range_site_primary_name
    - range_site_secondary_name
    - range_site_tertiary_name
  - Plant_Communities table
    - es_type
    - es_mlra
    - es_mlru
    - es_site_number
    - es_state
    - plant_community_id
    - plant_community_sequence
    - plant_community_name
    - state_id
  - Range_Annual_Production table
    - es_type
    - es_mlra
    - es_mlru
    - es_site_number

es_state
plant_community_id
plant_type
plnt_type_annual_production_rv

1.3. GIS Operations

For each AoA

If fis_method == 1 (FIS congruent with AoA boundary)

#Create FIS and associated attribute table

Use AoA geometry to create FIS geometry

Create attribute table (one record) with following attributes

- fis_id
- AoAId
- planner_id
- inventory_date
- fis_type … value is single
- fis_area

#Create FIS mapunit polygons (see Figure 1a-1)

Intersect FIS geometry with SSURGO mapunit geometry

Dissolve very small intersected polygons

temp_fis_mu_attrib table columns

- fis_id … one per AoA
- fis_area
- fis_type = 1 … single FIS per AoA
- AoAId
- planner_id
- inventory_date
- fis_mu_id … one or more per AoA
- mukey
- fis_mu_area

#Create FIS mapunit component table (see Table 1a-1)

Select

- temp_fis_mu_attrib.AoAId
- temp_fis_mu_attrib.planner_id
- temp_fis_mu_attrib.inventory_date
- temp_fis_mu_attrib.fis_id
- temp_fis_mu_attrib.fis_area
- temp_fis_mu_attrib.fis_mu_id
- temp_fis_mu_attrib.fis_mu_area
- component.mukey
- component.cokey
- component.compname
- component.otherph
- component.localphase

component.majcompflag
component.comppct_r
component.rsprod_r

Into temp_fis_mu_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO

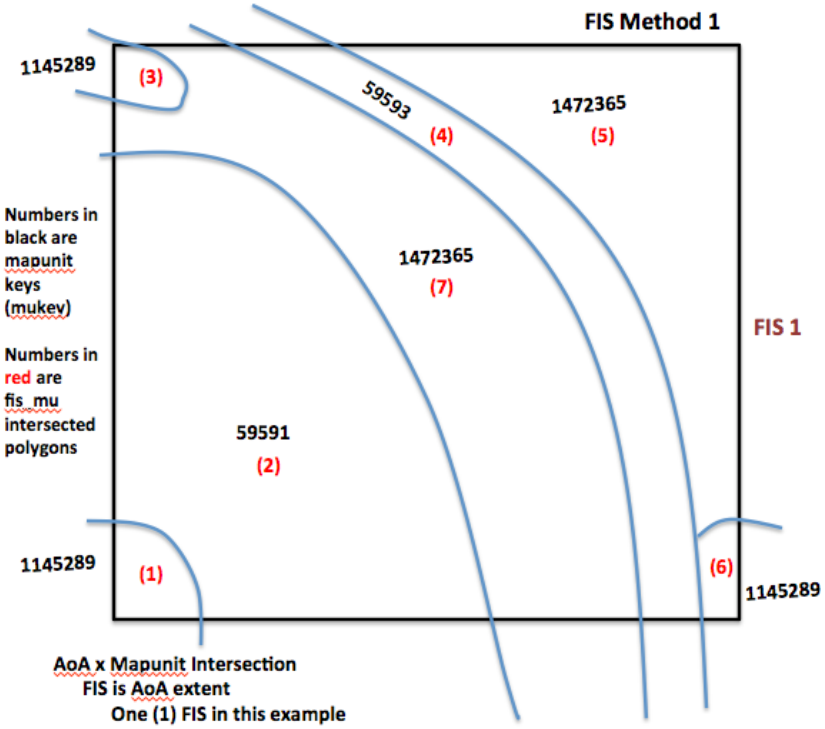


Figure 1a-1. FIS Method 1

fis_mu_id	fis_id	mukey	cokey	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area
1	1	1145289	11510284	200	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
2	1	59591	11510290	2500	Crossen			5	178.6
		59591	11510291		Altuda			5	178.6
		59591	11510292		Bissett			50	1785.7
		59591	11510294		Cienega			4	142.9
		59591	11510293		Blackgap			6	214.3
3	1	1145289	11510284	100	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
4	1	59593	11510303	850	Musgrave			10	85
		59593	11510304		Straddlebug			5	42.5
		59593	11510305		Borunda		gravelly	30	255
		59593	11510301		Borunda			50	425
		59593	11510302		Butcherknife			5	42.5
5	1	1472365	11510424	2000	Boracho			40	860.2
		1472365	11510425		Chilimol			40	860.2
		1472365	11510426		Berrend			13	279.6
6	1	1145289	11510284	100	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
7	1	1472365	11510424	2000	Boracho			40	860.2
		1472365	11510425		Chilimol			40	860.2
		1472365	11510426		Berrend			13	279.6

Table 1a-1. FIS Method 1 Mapunit Component Table

On temp_fis_mu_attrib.mukey=component.mukey
Order By temp_fis_mu_attrib.fis_mu_id

Alter Table temp_fis_mu_comp
Add (fis_mu_area * comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

Else if fis_method == 2 (FIS = fis_mu_id)

#Create one or more FIS within the AoA (See Figure 1a-2)
Intersect AoA and SSURGO mapunits to create one or more FIS polygons
Dissolve very small intersected polygons in the AoA

#Each intersected polygon is a FIS
temp_fis_mu_attrib table columns
- fis_id … one or more in the AoA
- fis_area
- AoAId … one per fis_id
- planner_id … one per fis_id
- inventory_date … one per fis_id
- fis_type = 2 … multiple FIS per AoA
- mukey … one associated with fis_id

#Create FIS mapunit component table (see Table 1a-2)
Select
temp_fis_mu_attrib.AoAId
temp_fis_mu_attrib.planner_id
temp_fis_mu_attrib.inventory_date
temp_fis_mu_attrib.fis_id
temp_fis_mu_attrib.fis_area
component.mukey
component.cokey
component.compname
component.otherph
component.localphase
component.majcompflag
component.comppct_r
component.rsprod_r
Into temp_fis_mu_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Order By temp_fis_mu_attrib.fis_id

Alter temp_Table fis_mu_comp
Add (fis_area* comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

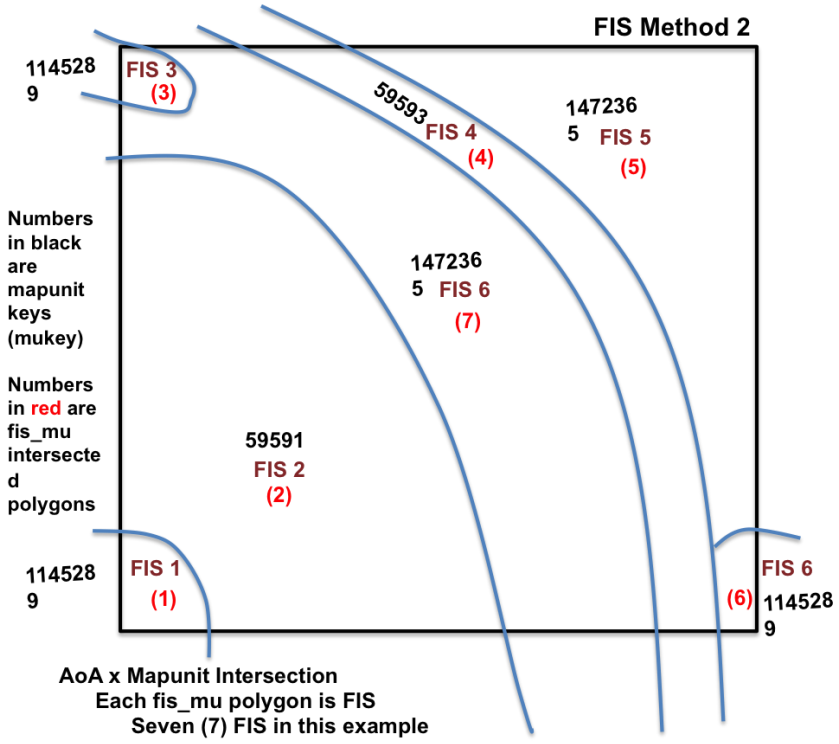


Figure 1a-2. FIS Method 2, Mapunit-Based Forage Inventory Sites

fis_mu_id	fis_id	mukey	cokey	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area
1	1	1145289	11510284	200	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
2	2	59591	11510290	2500	Crossen			5	178.6
		59591	11510291		Altuda			5	178.6
		59591	11510292		Bissett			50	1785.7
		59591	11510294		Cienega			4	142.9
		59591	11510293		Blackgap			6	214.3
3	3	1145289	11510284	100	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
4	4	59593	11510303	850	Musgrave			10	85
		59593	11510304		Straddlebug			5	42.5
		59593	11510305		Borunda		gravelly	30	255
		59593	11510301		Borunda			50	425
		59593	11510302		Butcherknife			5	42.5
5	5	1472365	11510424	2000	Boracho			40	860.2
		1472365	11510425		Chilimol			40	860.2
		1472365	11510426		Berrend			13	279.6
6	6	1145289	11510284	100	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
7	7	1472365	11510424	2000	Boracho			40	860.2
		1472365	11510425		Chilimol			40	860.2
		1472365	11510426		Berrend			13	279.6

Table 1a-2 FIS Method 2 Mapunit Component Table

Else if fis_method == 3 (FIS is AoA x user supplied geometry intersection boundary)

#Create FIS and associated attribute table

Copy AoA geometry to create FIS geometry

Create attribute table (one record) with following attributes

- fis_id
- AoAId
- planner_id
- inventory_date
- fis_type … value is single
- fis_area

#Create FIS mapunit polygons (see Figure 1a-3)

Intersect FIS geometry with SSURGO mapunit geometry

Dissolve very small intersected polygons

temp_fis_mu_attrib table columns

- fis_id … one or more per AoA
- fis_area
- fis_mu_id … one or more per FIS in the AoA

fis_mu_area
fis_type = 2 … multiple FIS per AoA
AoAId
planner_id
inventory_date
mukey

#Create FIS mapunit component table (see Table 1a-3)
Select
temp_fis_mu_attrib.AoAId
temp_fis_mu_attrib.planner_id
temp_fis_mu_attrib.inventory_date
temp_fis_mu_attrib.fis_id
temp_fis_mu_attrib.fis_area
temp_fis_mu_attrib.fis_mu_id
temp_fis_mu_attrib.fis_mu_area
component.mukey
component.cokey
component.compname
component.otherph
component.localphase
component.majcompflag
component.comppct_r
component.rsprod_r
Into temp_fis_mu_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Order By temp_fis_mu_attrib.fis_mu_id

Alter Table temp_fis_mu_comp
Add (fis_mu_area * comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

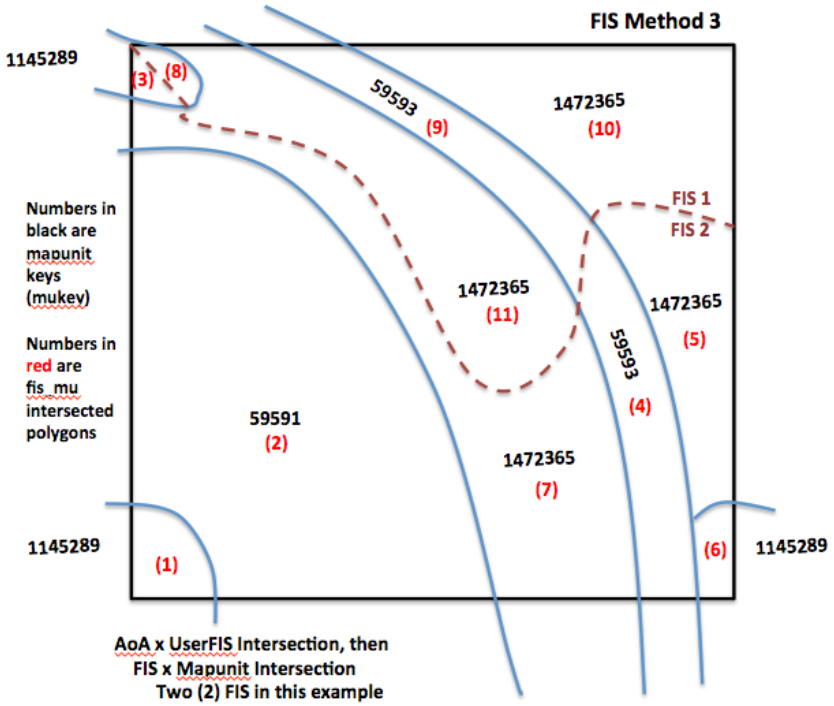


Figure 1a-3. FIS Method 3, User Defined Forage Inventory Sites

fis_mu_id	fis_id	mukey	cokey	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area
1	1	1145289	11510284	200	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
2	1	59591	11510290	2500	Crossen			5	178.6
		59591	11510291		Altuda			5	178.6
		59591	11510292		Bissett			50	1785.7
		59591	11510294		Cienega			4	142.9
		59591	11510293		Blackgap			6	214.3
3	1	1145289	11510284	50	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
4	1	59593	11510303	400	Musgrave			10	40
		59593	11510304		Straddlebug			5	20
		59593	11510305		Borunda		gravelly	30	120
		59593	11510301		Borunda			50	200
		59593	11510302		Butcherknife			5	20
5	1	1472365	11510424	1000	Boracho			40	430.1
		1472365	11510425		Chilimol			40	430.1
		1472365	11510426		Berrend			13	139.8
6	1	1145289	11510284	100	Berrend			72	161.8
		1145289	11510283		Espy			17	38.2
7	1	1472365	11510424	1500	Boracho			40	645.2
		1472365	11510425		Chilimol			40	645.2
		1472365	11510426		Berrend			13	209.7

Table 1a-3. FIS Method 3, Mapunit Component Table

fis_mu_id	fis_id	mukey	cokey	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area
8	2	1145289	11510284	50	Berrend			72	40.4
		1145289	11510283		Espy			17	9.6
9	2	59593	11510303	450	Musgrave			10	45
		59593	11510304		Straddlebug			5	22.5
		59593	11510305		Borunda		gravelly	30	135
		59593	11510301		Borunda			50	225
		59593	11510302		Butcherknife			5	22.5
10	2	1472365	11510424	1000	Boracho			40	430.1
		1472365	11510425		Chilimol			40	430.1
		1472365	11510426		Berrend			13	139.8
11	2	1472365	11510424	500	Boracho			40	215.1
		1472365	11510425		Chilimol			40	215.1
		1472365	11510426		Berrend			13	69.9

Table 1a-3. FIS Method 3, Mapunit Component Table (cont'd)

#Create set of FIS ecological sites and corresponding FIS percentage and highest SSURGO range production
Select
temp_fis_mu_attrib.AoAId
temp_fis_mu_attrib.planner_id
temp_fis_mu_attrib.inventory_date
temp_fis_mu_comp.fis_id
sum(temp_fis_mu_comp.fis_mu_comp_pctfis) **As** es_pctfis
max(temp_fis_mu_comp.rsprod_r **As** es_rsprod
coecoslass.ecoclassid
Into temp_fis_ecoclass table
From temp_fis_mu_comp table
Inner Join SSURGO ecoclassid table
On temp_fis_mu_comp.cokey=ecoclassid.cokey
Where ecoclassid LIKE 'R%' OR 'F%' and es_pctfis >= 5
Group By fis_id, ecoclassid

#Get ESIS ecological site identifiers and names for each ecoclassid (es_id)
Select
temp_fis_ecoclass.AoAId
temp_fis_ecoclass.planner_id
temp_fis_ecoclass.inventory_date
temp_fis_ecoclass.fis_id
temp_fis_ecoclass.es_pctfis
temp_fis_ecoclass.es_rsprod
ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state) **As** es_id
ecological_sites.concatenated(range_site_primary_name, range_site_secondary_name, range_site_tertiary_name) **As** es_range_name
ecological_sites.concatenated(forest_sitetree1_vernacular,

forest_sitetree2_vernacular, forest_siteshrub1_vernacular,
forest_siteshrub2_vernacular, forest_siteherb1_vernacular,
forest_siteherb2_vernacular) **As** es_forest_name
Into temp_es_list table
From ESIS ecological_sites table
Inner Join temp_fis_ecoclass
On ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=fis_ecoclass.ecoclassid

#Get one or more plant communities for each ecological site in the FIS
Select
temp_es_list.AoAId
temp_es_list.planner_id
temp_es_list.inventory_date
temp_es_list.es_id
temp_es_list.fis_id
temp_es_list.es_range_name
temp_es_list.es_forest_name
temp_es_list.es_pctfis
temp_es_list.es_rsprod
plant_communities.plant_community_id
plant_communities.plant_community_name
plant_communities.plant_community_sequence
plant_communities.state_id
Into temp_es_pc table
From ESIS plant_communities table
Inner Join temp_es_list table
On plant_communities.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=es_list.es_id

#Get ecological state for each plant community of each ecological site in the FIS
Select
temp_es_pc.AoAId
temp_es_pc.planner_id
temp_es_pc.inventory_date
temp_es_pc.es_id
temp_es_pc.fis_id
temp_es_pc.es_range_name
temp_es_pc.es_forest_name
temp_es_pc.es_pctfis
temp_es_pc.es_rsprod
temp_es_pc.plant_community_id
temp_es_pc.plant_community_name
temp_es_pc.plant_communty_sequence
temp_es_pc.state_id
ecological_site_state.state_name
Into temp_es_pc_state

From ESIS ecological_site_state table
Inner Join temp_es_pc table
On ecological_site_state.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_es_pc.es_id

#Get estimated production for each plant community of each ecological site in the FIS
Select
temp_es_pc_state.AoAId
temp_es_pc_ state.planner_id
temp_es_pc_ state.inventory_date
temp_es_pc_state.es_id
temp_es_pc_state.fis_id
temp_es_pc_state.es_range_name
temp_es_pc_state.es_forest_name
temp_es_pc_state.es_pctfis
temp_es_pc_state.es_rsprod
temp_es_pc_state.plant_community_id
temp_es_pc_state.plant_community_name
temp_es_pc_state.plant_communty_sequence
temp_es_pc_state.state_id
temp_es_pc_state.state_name
range_annual_production.plant_community_id
sum(range_annual_production.plnt_type_annual_production_rv) **As** es_est_prod
Into temp_es_est_prod
From ESIS range_annual_production table
Inner Join temp_es_pc_state table
On range_annual_production.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_es_pc_state.es_id and range_annual_production.plant_community_id= temp_es_pc_state.plant_community_id

#Send to Output
Output contents of temp_es_est_prod table

#The data in es_est_prod should enable the application to create the following choice lists, depending on the method for creating forage inventory sites.

Method 1 – AoA-Extent FIS

FIS 1

Ecological Site	% of FIS	Ecological State	Community Phase	ESD Production	SSURGO Production	Select
GRAS es_name	calc	ESIS state_name	ESIS plant_community_name	calc	rsprod_r	
Shallow, Mixed Prairie	32	Native Grassland	Gramas/Mixed Shrubs	900		☐
			Fluffgrass-Gramas, Mixed Shrubs			☐
		Shrubland	Mixed Shrubs/Gramas			☐
			Mixed Shrubs/Fluffgrass-Gramas			☐
		Non-Native Grassland	Lehmann lovegrass-Grama/Mixed Shrubs			☐
		Non-Native Shrubland	Mixed Shrubs - Lehman lovegrass			☐
Gravelly, Mixed Prairie	32	Native Grassland	Grama Dominant Community	1100		☐
			Patchy Grama Community	900		☐
Loamy Slope, Mixed Prairie	19	NA	NA	NA	1500	☐
Loamy, Desert Grassland	11	NA	NA	NA	1000	☐
Gravelly, Desert Grassland	7	Native Grassland	Short & Midgrass/Shrub Complex	660		☐
			Shrub/Shortgrass Complex	300		☐

Method 2 – Mapunit-Based FIS

FIS 1 | FIS 2 | FIS 3 | FIS 4 | FIS 5 | FIS 6 | FIS 7

Ecological Site	% of FIS	Ecological State	Community Phase	ESD Production	SSURGO Production	Select
GRAS es_name	calc	ESIS state_name	ESIS plant_community_name	calc	rsprod_r	
Loamy Slope, Mixed Prairie	81	NA	NA	NA	1500	☑
Shallow, Mixed Prairie	19	Native Grassland	Gramas/Mixed Shrubs	900		☐
			Fluffgrass-Gramas, Mixed Shrubs			☐
		Shrubland	Mixed Shrubs/Gramas			☐
			Mixed Shrubs/Fluffgrass-Gramas			☐
		Non-Native Grassland	Lehmann lovegrass-Grama/Mixed Shrubs			☐
		Non-Native Shrubland	Mixed Shrubs - Lehman lovegrass			☐

Method 3 – User Delineated FIS

FIS 1 | FIS 2

Ecological Site	% of FIS	Ecological State	Community Phase	ESD Production	SSURGO Production	Select
GRAS es_name	calc	ESIS state_name	ESIS plant_community_name	calc	rsprod_r	
Shallow, Mixed Prairie	32	Native Grassland	Gramas/Mixed Shrubs	900		☐
			Fluffgrass-Gramas, Mixed Shrubs			☐
		Shrubland	Mixed Shrubs/Gramas			☐
			Mixed Shrubs/Fluffgrass-Gramas			☐
		Non-Native Grassland	Lehmann lovegrass-Grama/Mixed Shrubs			☐
		Non-Native Shrubland	Mixed Shrubs - Lehman lovegrass			☐
Gravelly, Mixed Prairie	32	Native Grassland	Grama Dominant Community	1100		☐
			Patchy Grama Community	900		☐
Loamy Slope, Mixed Prairie	19	NA	NA	NA	1500	☐
Loamy, Desert Grassland	11	NA	NA	NA	1000	☐
Gravelly, Desert Grassland	7	Native Grassland	Short & Midgrass/Shrub Complex	660		☐
			Shrub/Shortgrass Complex	300		☐

1.1. Output

AoAId … one

- planner_id
- inventory_date
- fis_type
- fis_id
  - fis_geometry
  - es_id
    - es_range_name
    - es_forest_name
    - es_pctfis
    - es_rsprod
    - plant_community_id
      - plant_community_name
      - plant_communty_sequence
      - state_id
      - state_name
      - es_est_prod

Service GRAS-1b: Create Forage Inventory Sites, Get Forage Suitability Groups, Their Plant Species, and Estimated Production for an Area of Analysis (CreateFSGFIS)

Purpose: Create forage inventory sites (FISs) for an area of analysis (AoA) and get estimated production from associated Ecological Site Information System (ESIS) forage suitability group (FSG) species or associated SSURGO soil components or mapunits.

This service is similar to GRAS-1a except it supports get estimated production method 2 – From Forage Suitability Groups. FSGs are similar to ecological sites except they usually apply to AoAs having a land use of pasture. Pasture AoAs usually contain seeded plant species, alone or in relatively simple mixtures, and are managed more like crops, whether irrigated, fertilized, or otherwise treated to optimize production.

The six methods (abstracted to three) in GRAS-1b for creating FIS polygons apply to this service. Any requires the creation of a table containing soil components relevant to the FIS. Soil components can link to FSGs in ESIS containing estimated dry matter production for their associated plant species group. SSURGO components themselves also can contain dry matter production values. Associated soil mapunits also may contain dry matter production values, and serve as a backup if no production exists in ESIS for FSGs.

This service consumes an application request payload to create one or more FIS per AoA, get associated FSGs, and get estimated dry matter production for associated plant species groups and soil components. The results payload returns data enabling the requesting application to populate a choice list for associating estimated dry matter production to a FIS or FIS component.

Service Signature

Request Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
- aoa_geometry … geospatial coordinates, one set per AoA, Area of Analysis Polygon Geometry
- fis_method … integer, Method for Creating Forage Inventory Site; choices are 1 - FIS is AoA boundary, 2 – FIS is AoA x mapunit intersection boundary, 3 – FIS is AoA x user supplied geometry intersection boundary
- est_prod_method … integer, value is 2 for this service; Method to Get Estimated Forage Production; see GRAS 1a for method 1 – from ESIS ecological site; for this service: method 2 – from ESIS forage suitability group, and GRAS-1c for method 3 – from SSURGO
- user_fis_geometry … geospatial coordinates, User Supplied Geometry for Creating Forage Inventory Sites; can be one or more lines or polygons
- planner_id … character varying(23), Application User Identifier
- inventory_date … Date (yyyy-mm-dd), Forage Inventory Date

Result Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
- planner_id … character varying(23), Application User Identifier
- inventory_date … Date (yyyy-mm-dd), Forage Inventory Date

fis_type … integer, Forage Inventory Site Type; values are 1 – Single or 2 - Multiple
fis_id … one or more per AoA depending on FIS method, Forage Inventory Site Identifier
fis_geometry … geospatial coordinates, one or more sets for the AoA, Forage Inventory Site Geometry
fsg_id … character varying(60), one or more per FIS, Forage Suitability Group Identifier
fsg_name … character varying(120), Forage Suitability Group Name, name applied if ecological site identifier begins with G
fsg_pctfis
fsg_production_id … one or more per FSG
fsg_species_list … double precision, Percent of Forage Inventory Site
fsg_production_type … integer, Forage Production Type; values are 1 – Forage or 2 – Pasture
dry_aum_high … bigint, Estimated Dryland Animal Unit Month Production
irrig_aum_high … bigint, Estimated Irrigated Animal Unit Month Production
dry_lbs_high … bigint, Estimated Dryland Pounds Per Acre Production
irrig_lbs_high ... bigint, Estimated Irrigated Pounds Per Acre Production
cocropyldkey … character varying(60), one or more per Forage Suitability Group, Soil Component Crop Yield Key
cropname … character varying(508), Crop Name
nonirryield_aum … numeric, Non-Irrigated Yield in AUMs
irryield_aum … numeric, Irrigated Yield in AUMs
mucropyldkey … character varying(60), one or more per Forage Suitability Group, Mapunit Crop Yield Key
cropname … character varying(508), Crop Name
nonirryield_aum … numeric, Non-Irrigated Yield in AUMs
irryield_aum … numeric, Irrigated Yield in AUMs

Reference Data Sources

SSURGO Data Mart
mapunit table
component table
coecoclass table

ESIS Data Mart
Ecological_Sites table
FSG_Production table
FSG_Species_Production_Group table
Plant_Master table

Component

1. <u>Create Forage Inventory Sites and Get FSG-Based Estimated Production (FISFSGProd)</u>

1.1. Inputs

AoA identifier
AoA polygon geometry
fis_method
User FIS geometry … if FIS method 3
planner_id
inventory_date

1.2. Data

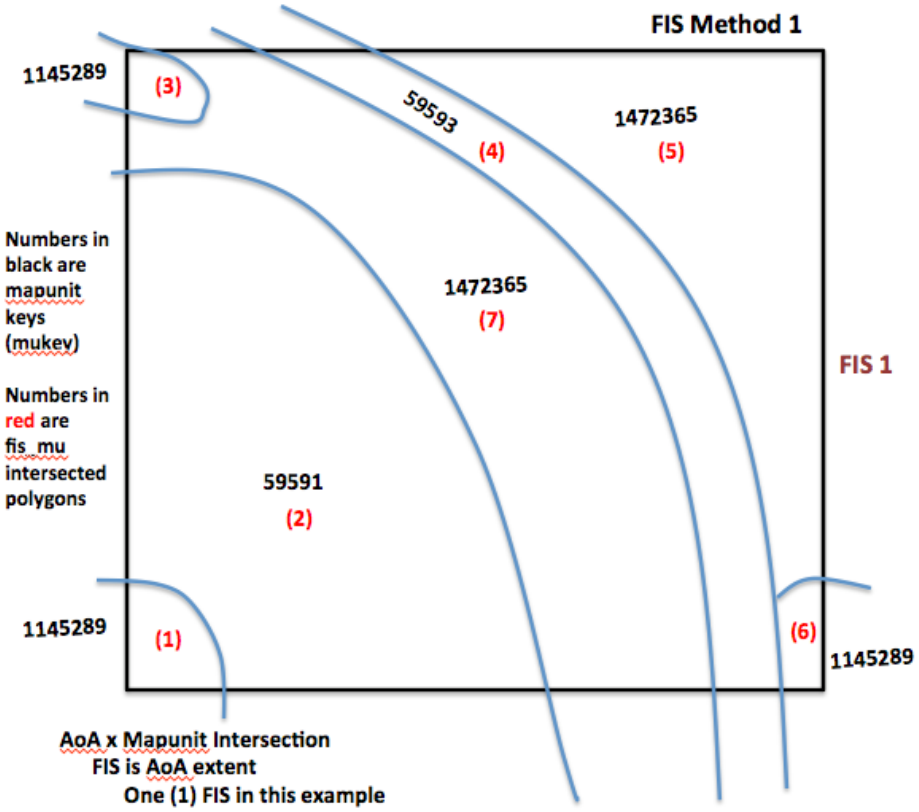
SSURGO
mapunit
mukey
component
mukey
cokey
majcompflag
comppct_r

1.3. GIS Operations

For each AoA
If fis_method == 1 (FIS congruent with AoA boundary)
#Create FIS and associated attribute table
Copy AoA geometry to create FIS geometry
Create attribute table (one record) with following attributes
fis_id
AoAId
planner_id
inventory_date
fis_type … value is single
fis_area

#Create FIS mapunit polygons (see following figure)
Intersect FIS geometry with SSURGO mapunit geometry
Dissolve very small intersected polygons

temp_fis_mu_attrib table columns
fis_id … one per AoA
fis_area
fis_type = 1 … single FIS per AoA
AoAId
planner_id
inventory_date
fis_mu_id … one or more per AoA
mukey
fis_mu_area



#Create FIS mapunit component table (see following table)
Select
temp_fis_mu_attrib.AoAId
temp_fis_mu_attrib.planner_id
temp_fis_mu_attrib.inventory_date
temp_fis_mu_attrib.fis_id
temp_fis_mu_attrib.fis_area
temp_fis_mu_attrib.fis_mu_id
temp_fis_mu_attrib.fis_mu_area
component.mukey
component.cokey
component.compname
component.otherph
component.localphase
component.majcompflag
component.comppct_r
coecoclass.ecoclassid
Into temp_fis_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Inner Join coecoclass table in SSURGO
On component.cokey=coecoclass.cokey

Where coecoclass.ecoclassid LIKE 'G%' and coecoclass.ecoclasstypename = 'Forage Suitability Groups'
Order By temp_fis_mu_attrib.fis_mu_id

Alter Table temp_fis_comp
Add (fis_mu_area * comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

#Create set of FIS forage suitability groups and corresponding FIS percentage
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
sum(temp_fis_comp.fis_mu_comp_pctfis) **As** fsg_pctfis
temp_fis_comp.ecoclassid
Into temp_fis_fsg table
From temp_fis_comp table
Where fsg_pctfis >= 5
Group By fis_id, ecoclassid

#Get ESIS ecological site identifiers and names for each forage suitability group
Select
temp_fis_fsg.AoAId
temp_fis_fsg.planner_id
temp_fis_fsg.inventory_date
temp_fis_fsg.fis_id
temp_fis_fsg.fsg_pctfis
ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state) **As** fsg_id
ecological_sites.range_site_primary_name **As** fsg_name
Into temp_fsg table
From ESIS ecological_sites table
Inner Join temp_fis_fsg table
On ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_fis_fsg.ecoclassid

fis_mu_id	fis_id	mukey	cokey	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area
1	1	2581164	11328534	200	Hamar			71	142.0
		2581164	11328530		Garborg			13	26.0
		2581164	11328533		Hecla			8	16.0
		2581164	11328531		Ulen			5	10.0
		2581164	11328532		Wyndmere			3	6.0
2	1	2581147	11328413	2500	Sioux			64	1600.0
		2581147	11328415		Renshaw			26	650.0
		2581147	11328414		Fordville			4	100.0
		2581147	11328412		Arvilla			4	100.0
		2581147	11328411		Divide			2	50.0
3	1	2581164	11328534	100	Hamar			71	71.0
		2581164	11328530		Garborg			13	13.0
		2581164	11328533		Hecla			8	8.0
		2581164	11328531		Ulen			5	5.0
		2581164	11328532		Wyndmere			3	3.0
4	1	2581114	11328201	850	Cresbard			49	416.5
		2581114	11328200		Cavour			25	212.5
		2581114	11328196		Barnes			9	76.5
		2581114	11328199		Hamerly		moderately saline	5	42.5
		2581114	11328198		Ferney			5	42.5
		2581114	11328197		Svea		saline-sodic substratum	3	25.5
		2581114	11328202		Tonka			2	17.0
		2581114	11328203		Vallers		moderately saline	2	17.0
5	1	2581177	11328579	2000	Eckman			53	1060.0
		2581177	11328577		Zell			20	400.0
		2581177	11328576		Gardena			13	260.0
		2581177	11328578		Barnes			4	80.0
		2581177	11328580		Glyndon			4	80.0
		2581177	11328582		Tonka			3	60.0
		2581177	11328581		Cresbard			3	60.0
6	1	2581164	11328534	100	Hamar			71	71.0
		2581164	11328530		Garborg			13	13.0
		2581164	11328533		Hecla			8	8.0
		2581164	11328531		Ulen			5	5.0
		2581164	11328532		Wyndmere			3	3.0
7	1	2581177	11328579	2000	Eckman			53	1060.0
		2581177	11328577		Zell			20	400.0
		2581177	11328576		Gardena			13	260.0
		2581177	11328578		Barnes			4	80.0
		2581177	11328580		Glyndon			4	80.0
		2581177	11328582		Tonka			3	60.0
		2581177	11328581		Cresbard			3	60.0

#Get estimated production for each species or species mixture of the forage suitability group in the FIS
Select
temp_fsg.AoAId
temp_fsg.planner_id
temp_fsg.inventory_date
temp_fsg.fsg_id
temp_fsg.fsg_name
temp_fsg.fis_id
temp_fsg.fsg_pctfis
fsg_production.fsg_production_id
fsg_production.fsg_productiong_type
fsg_production.dry_aum_high
fsg_production.irrig_aum_high
fsg_production.dry_lbs_high
fsg_production.irrig_lbs_high
Into temp_fsg_est_prod
From ESIS fsg_production table

Inner Join temp_fsg table
On fsg_production.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_fsg.fsg_id

#Get plant symbols for each FSG production group
Select
    temp_fsg est_prod.AoAId
    temp_fsg est_prod.planner_id
    temp_fsg est_prod.inventory_date
    temp_fsg_est_prod.fsg_id
    temp_fsg_est_prod.fsg_name
    temp_fsg_est_prod.fis_id
    temp_fsg_est_prod.fsg_pctfis
    temp_fsg_est_prod.fsg_production_id
    temp_fsg_est_prod.fsg_production_type
    temp_fsg_est_prod.dry_aum_high
    temp_fsg_est_prod.irrig_aum_high
    temp_fsg_est_prod.dry_lbs_high
    temp_fsg_est_prod.irrig_lbs_high
    fsg_species_production_group.plant_symbol
Into temp_fsg_est_prod2
From ESIS fsg_species_production_group table
Inner Join temp_fsg_est_prod table
On fsg_species_production_group concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_fsg_est_prod.fsg_id and fsg_species_production_group.fsg_production_id= temp_fsg_est_prod.fsg_production_id

#Get plant species names for FSG plant symbols
Select
    temp_fsg_est_prod2.fsg_id
    temp_fsg_est_prod2.fis_id
    temp_fsg_est_prod2.fsg_production_id
    plant_master.vernacular
Into temp_fsg_species
From plant_master
Inner Join temp_fsg_est_prod2 **On** plant_master.symbol= temp_fsg_est_prod2.plant_symbol

#Following probably is not correct, but plant species names for a fsg production group must be concantenated into a single data field
Select
    temp_fsg_species.fsg_id
    temp_fsg_species.fis_id
    temp_fsg_species.fsg_production_id
    Stuff((
        Select ',' + temp_fsg_species.vernacular **As** [text()]

```
        #Add a comma (,) before each value
        From temp_fsg_species
        #Select it as XML and remove 1st character (,) from result
        For XML PATH('') ), 1, 1, '' )
        As temp_fsg_species_list
Into temp_fsg_species2
From temp_fsg_species

#Complete FSG estimated production table
Select
    temp_fsg est_prod2.AoAId
    temp_fsg est_prod2.planner_id
    temp_fsg est_prod2.inventory_date
    temp_fsg_est_prod2.fsg_id
    temp_fsg_est_prod2.fsg_name
    temp_fsg_est_prod2.fis_id
    temp_fsg_est_prod2.fsg_pctfis
    temp_fsg_est_prod2.fsg_production_id
    temp_fsg_species2.fsg_species_list
    temp_fsg_est_prod2.fsg_production_type
    temp_fsg_est_prod2.dry_aum_high
    temp_fsg_est_prod2.irrig_aum_high
    temp_fsg_est_prod2.dry_lbs_high
    temp_fsg_est_prod2.irrig_lbs_high
Into temp_fsg_est_prod3
From temp_fsg_est_prod2
Inner Join temp_fsg_species2
On temp_fsg_est_prod2.fsg_id= temp_fsg_species2.fsg_id and
temp_fsg_est_prod2.fis_id= temp_fsg_species2.fis_id and
temp_fsg_est_prod2.fsg_production_id=
temp_fsg_species2.fsg_production_id
Group By fsg_species_list, fsg_production_id

#Send to Output
Output contents of temp_fsg_est_prod3 table

#Create list of soil components for each FSG in the FIS
Select
    temp_fis_comp.fis_id
    temp_fis_comp.cokey
    temp_fis_comp.compname
    temp_fis_comp.ecoclassid As fsg_id
Into temp_fsg_comp
From temp_fis_comp
Group By ecoclassid, compname

#Get maximum irrigated and non-irrigated forage production from SSURGO
```

for each soil component crop represented in the FIS
Select
temp_fsg_comp.fsg_id
temp_fsg_comp.fis_id
temp_fsg_comp.cokey
cocropyld.cocropyldkey
cocropyld.cropname
max(cocropyld.nonirryield_r) **As** nonirryld_aum
max(cocropyld.irryield_r) **As** irryld_aum
Into temp_fsg_comp_est_prod
From SSURGO cocropyld table
Inner Join temp_fsg_comp table **On** cocropyld.cokey= temp_fsg_comp.cokey
Where cocropyld.yldunits=AUM
Group By fsg_id, cropname

#Send to Output
Output data in temp_fsg_comp_est_production

#Create list of soil mapunits for each FSG in the FIS
Select
temp_fis_comp.fis_id
temp_fis_comp.mukey
temp_fis_comp.muname
temp_fis_comp.ecoclassid **As** fsg_id
Into temp_fsg_comp
From temp_fis_comp
Group By ecoclassid, muname

#Get maximum irrigated and non-irrigated forage production from SSURGO for each soil mapunit crop represented in the FIS
Select
temp_fsg_comp.fsg_id
temp_fsg_comp.fis_id
temp_fsg_comp.mukey
mucropyld.mucropyldkey
mucropyld.cropname
max(mucropyld.nonirryield_r) **As** nonirryld_aum
max(mucropyld.irryield_r) **As** irryld_aum
Into temp_fsg_mu_est_prod
From SSURGO mucropyld table
Inner Join temp_fsg_comp table **On** mucropyld.cokey= temp_fsg_comp.cokey
Where mucropyld.yldunits=AUM
Group By fsg_id, cropname

#Send to Output
Output data in temp_fsg_mu_est_production

#The data in fsg_est_prod3, fsg_comp_est_prod, and fsg_mu_est_prod should enable the application to create the following choice list

FIS 1 Forage Suitability Group	% of FIS	Producton Type	Plant Species	FSG Dry (lbs/ac)	FSG Dry (lbs/ac)	FSG Dry (AUM)	FSG Irrig (AUM)	SSURGO Dry (AUM)	SSURGO Irrig (AUM)	Select
fsg_name	calc									
Loam	31	Forage	Alfalfa	3900						☑
			Alfalfa, Intermediate Wheatgrass	3500						☐
			Smooth Brome, Alfalfa	3500						☐
			Big Bluestem	2900						☐
			Crested Wheat Grass	2600						☐
			Green Needlegrass	2000						☐
			Intermediate Wheatgrass	2800						☐
			Smooth Brome	2800						☐
			Switchgrass	3100						☐
			Western Wheatgrass	2000						☐
		Pasture	(None)							☐
Wet	26	Forage	(None)							☐
		Pasture	(None)							☐
		Soil Component	Tall Wheatgrass					3.7		☐
			Bromegrass-Alfalfa					3.8		☐
Limy Upland	10	Forage	Crested Wheat Grass, Alfalfa	2400						☐
			Alfalfa, Intermediate Wheatgrass	2700						☐
			Crested Wheat Grass	2400						☐
			Intermediate Wheatgrass	2500						☐
			Little Bluestem	2300						☐
		Pasture	(None)							☐
Very Droughty Loam	10	Forage	Crested Wheat Grass	1400						☐
			Intermediate Wheatgrass	1700						☐
			Western Wheatgrass	1100						☐
		Pasture	(None)							☐
Overflow	7	Forage	Alfalfa	5400						☐
			Alfalfa, Intermediate Wheatgrass	4000						☐
			Smooth Brome, Alfalfa	4000						☐
			Big Bluestem	3700						☐
			Indiangrass	3100						☐
			Intermediate Wheatgrass	3100						☐
			Smooth Brome	3100						☐
			Switchgrass	4000						☐
		Pasture	(None)							☐
Clayey Subsoil	6	Forage	Alfalfa	3400						☐
			Alfalfa, Intermediate Wheatgrass	3200						☐
			Smooth Brome, Alfalfa	3200						☐
			Big Bluestem	2600						☐
			Crested Wheat Grass	2500						☐
			Green Needlegrass	1600						☐
			Intermediate Wheatgrass	2300						☐
			Smooth Brome	2300						☐
			Switchgrass	2900						☐
			Western Wheatgrass	1900						☐
		Pasture	(None)							☐

Else if fis_method == 2 (FIS = fis_mu_id)

#Create one or more FIS within the AoA

Intersect AoA and SSURGO mapunits to create one or more FIS polygons

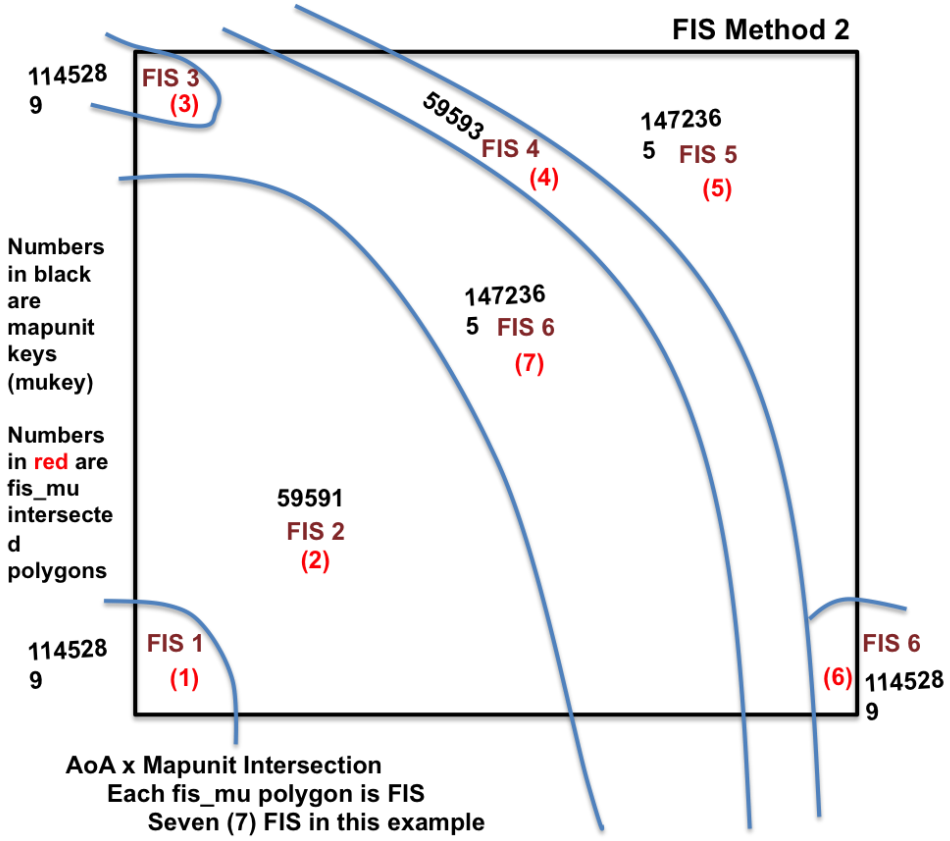
Dissolve very small intersected polygons in the AoA

#Each intersected polygon is a FIS

temp_fis_mu_attrib table columns

fis_id … one or more in the AoA

fis_area
AoAId … one per fis_id
planner_id … one per fis_id
inventory_date … one per fis_id
fis_type = 2 … multiple FIS per AoA
mukey … one associated with fis_id



#Create FIS mapunit component table (see following table)
Select
temp_fis_mu_attrib.AoAId
temp_fis_mu_attrib.planner_id
temp_fis_mu_attrib.inventory_date
temp_fis_mu_attrib.fis_id
temp_fis_mu_attrib.fis_area

component.mukey
component.cokey
component.compname
component.otherph
component.localphase
component.majcompflag
component.comppct_r

Into temp_fis_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Order By temp_fis_mu_attrib.fis_id

Alter Table temp_fis_comp
Add (fis_area* comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

fis_mu_id	fis_id	mukey	cokey	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area
1	1	2581164	11328534	200	Hamar			71	142.0
		2581164	11328530		Garborg			13	26.0
		2581164	11328533		Hecla			8	16.0
		2581164	11328531		Ulen			5	10.0
		2581164	11328532		Wyndmere			3	6.0
2	2	2581147	11328413	2500	Sioux			64	1600.0
		2581147	11328415		Renshaw			26	650.0
		2581147	11328414		Fordville			4	100.0
		2581147	11328412		Arvilla			4	100.0
		2581147	11328411		Divide			2	50.0
3	3	2581164	11328534	100	Hamar			71	71.0
		2581164	11328530		Garborg			13	13.0
		2581164	11328533		Hecla			8	8.0
		2581164	11328531		Ulen			5	5.0
		2581164	11328532		Wyndmere			3	3.0
4	4	2581114	11328201	850	Cresbard			49	416.5
		2581114	11328200		Cavour			25	212.5
		2581114	11328196		Barnes			9	76.5
		2581114	11328199		Hamerly		moderately saline	5	42.5
		2581114	11328198		Ferney			5	42.5
		2581114	11328197		Svea		saline-sodic substratum	3	25.5
		2581114	11328202		Tonka			2	17.0
		2581114	11328203		Vallers		moderately saline	2	17.0
5	5	2581177	11328579	2000	Eckman			53	1060.0
		2581177	11328577		Zell			20	400.0
		2581177	11328576		Gardena			13	260.0
		2581177	11328578		Barnes			4	80.0
		2581177	11328580		Glyndon			4	80.0
		2581177	11328582		Tonka			3	60.0
		2581177	11328581		Cresbard			3	60.0
6	6	2581164	11328534	100	Hamar			71	71.0
		2581164	11328530		Garborg			13	13.0
		2581164	11328533		Hecla			8	8.0
		2581164	11328531		Ulen			5	5.0
		2581164	11328532		Wyndmere			3	3.0
7	7	2581177	11328579	2000	Eckman			53	1060.0
		2581177	11328577		Zell			20	400.0
		2581177	11328576		Gardena			13	260.0
		2581177	11328578		Barnes			4	80.0
		2581177	11328580		Glyndon			4	80.0
		2581177	11328582		Tonka			3	60.0
		2581177	11328581		Cresbard			3	60.0

#Create set of FIS forage suitability groups and corresponding FIS percentage
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
sum(temp_fis_comp.fis_mu_comp_pctfis) **As** fsg_pctfis

coecoclass.ecoclassid
Into temp_fis_fsg table
From temp_fis_comp table
Inner Join SSURGO coecoclass table
On temp_fis_comp.cokey=coecoclassid.cokey
Where fsg_pctfis >= 5
Group By fis_id, ecoclassid

#Get ESIS ecological site identifiers and names for each forage suitability group
Select
temp_fis_fsg.AoAId
temp_fis_fsg.planner_id
temp_fis_fsg.inventory_date
temp_fis_fsg.fis_id
temp_fis_fsg.fsg_pctfis
ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state) **As** fsg_id
ecological_sites.range_site_primary_name **As** fsg_name
Into temp_fsg table
From ESIS ecological_sites table
Inner Join temp_fis_fsg table
On ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_fis_fsg.ecoclassid

#Get estimated production for each species or species mixture of the forage suitability group in the FIS
Select
temp_fsg.AoAId
temp_fsg.planner_id
temp_fsg.inventory_date
temp_fsg.fsg_id
temp_fsg.fsg_name
temp_fsg.fis_id
temp_fsg.fsg_pctfis
fsg_production.fsg_production_id
fsg_production.fsg_productiong_type
fsg_production.dry_aum_high
fsg_production.irrig_aum_high
fsg_production.dry_lbs_high
fsg_production.irrig_lbs_high
Into temp_fsg_est_prod
From ESIS fsg_production table
Inner Join temp_fsg table
On fsg_production.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_fsg.fsg_id

```
#Get plant symbols for each FSG production group
Select
    temp_fsg est_prod.AoAId
    temp_fsg est_prod.planner_id
    temp_fsg est_prod.inventory_date
    temp_fsg_est_prod.fsg_id
    temp_fsg_est_prod.fsg_name
    temp_fsg_est_prod.fis_id
    temp_fsg_est_prod.fsg_pctfis
    temp_fsg_est_prod.fsg_production_id
    temp_fsg_est_prod.fsg_production_type
    temp_fsg_est_prod.dry_aum_high
    temp_fsg_est_prod.irrig_aum_high
    temp_fsg_est_prod.dry_lbs_high
    temp_fsg_est_prod.irrig_lbs_high
    fsg_species_production_group.plant_symbol
Into temp_fsg_est_prod2
From ESIS fsg_species_production_group table
Inner Join temp_fsg_est_prod table
On fsg_species_production_group concatenated(es_type, es_mlra, es_mlru,
es_site_number, es_state)= temp_fsg_est_prod.fsg_id and
fsg_species_production_group.fsg_production_id=
temp_fsg_est_prod.fsg_production_id

#Get plant species names for FSG plant symbols
Select
    temp_fsg_est_prod2.fsg_id
    temp_fsg_est_prod2.fis_id
    temp_fsg_est_prod2.fsg_production_id
    plant_master.vernacular
Into temp_fsg_species
From plant_master
Inner Join temp_fsg_est_prod2 On plant_master.symbol=
temp_fsg_est_prod2.plant_symbol

#Following probably is not correct, but plant species names for a fsg
production group must be concantenated into a single data field
Select
    temp_fsg_species.fsg_id
    temp_fsg_species.fis_id
    temp_fsg_species.fsg_production_id
    Stuff((
        Select ',' + temp_fsg_species.vernacular As [text()]
        #Add a comma (,) before each value
        From temp_fsg_species
        #Select it as XML and remove 1st character (,) from result
```

```
        For XML PATH('') ), 1, 1, '' )
        As temp_fsg_species_list
Into temp_fsg_species2
From temp_fsg_species

#Complete FSG estimated production table
Select
    temp_fsg est_prod2.AoAId
    temp_fsg est_prod2.planner_id
    temp_fsg est_prod2.inventory_date
    temp_fsg_est_prod2.fsg_id
    temp_fsg_est_prod2.fsg_name
    temp_fsg_est_prod2.fis_id
    temp_fsg_est_prod2.fsg_pctfis
    temp_fsg_est_prod2.fsg_production_id
    temp_fsg_species2.fsg_species_list
    temp_fsg_est_prod2.fsg_production_type
    temp_fsg_est_prod2.dry_aum_high
    temp_fsg_est_prod2.irrig_aum_high
    temp_fsg_est_prod2.dry_lbs_high
    temp_fsg_est_prod2.irrig_lbs_high
Into temp_fsg_est_prod3
From temp_fsg_est_prod2
Inner Join temp_fsg_species2
On temp_fsg_est_prod2.fsg_id= temp_fsg_species2.fsg_id and
temp_fsg_est_prod2.fis_id= temp_fsg_species2.fis_id and
temp_fsg_est_prod2.fsg_production_id=
temp_fsg_species2.fsg_production_id
Group By fsg_species_list, fsg_production_id

#Send to Output
Output contents of temp_fsg_est_prod3 table

#Create list of soil components for each FSG in the FIS
Select
    temp_fis_comp.fis_id
    temp_fis_comp.cokey
    temp_fis_comp.compname
    temp_fis_comp.ecoclassid As fsg_id
Into temp_fsg_comp
From temp_fis_comp
Group By ecoclassid, compname

#Get maximum irrigated and non-irrigated forage production from SSURGO
for each soil component crop represented in the FIS
Select
    temp_fsg_comp.fsg_id
```

```
    temp_fsg_comp.fis_id
    temp_fsg_comp.cokey
    cocropyld.cocropyldkey
    cocropyld.cropname
    max(cocropyld.nonirryield_r) As nonirryld_aum
    max(cocropyld.irryield_r) As irryld_aum
Into temp_fsg_comp_est_prod
From SSURGO cocropyld table
Inner Join temp_fsg_comp table On cocropyld.cokey= temp_fsg_comp.cokey
Where cocropyld.yldunits=AUM
Group By fsg_id, cropname

#Send to Output
Output data in temp_fsg_comp_est_production

#Create list of soil mapunits for each FSG in the FIS
Select
    temp_fis_comp.fis_id
    temp_fis_comp.mukey
    temp_fis_comp.muname
    temp_fis_comp.ecoclassid As fsg_id
Into temp_fsg_comp
From temp_fis_comp
Group By ecoclassid, muname

#Get maximum irrigated and non-irrigated forage production from SSURGO
for each soil mapunit crop represented in the FIS
Select
    temp_fsg_comp.fsg_id
    temp_fsg_comp.fis_id
    temp_fsg_comp.mukey
    mucropyld.mucropyldkey
    mucropyld.cropname
    max(mucropyld.nonirryield_r) As nonirryld_aum
    max(mucropyld.irryield_r) As irryld_aum
Into temp_fsg_mu_est_prod
From SSURGO mucropyld table
Inner Join temp_fsg_comp table On mucropyld.cokey= temp_fsg_comp.cokey
Where mucropyld.yldunits=AUM
Group By fsg_id, cropname

#Send to Output
Output data in temp_fsg_mu_est_production

#The data in es_est_prod should enable the application to create the following choice list
```

FIS 1	FIS 2		FIS 3	FIS 4	FIS 4		FIS 6		FIS 7	
Forage Suitability Group	% of FIS	Producton Type	Plant Species	FSG Dry (lbs/ac)	FSG Dry (lbs/ac)	FSG Dry (AUM)	FSG Irrig (AUM)	SSURGO Dry (AUM)	SSURGO Irrig (AUM)	Select
fsg_name	calc									
Not Suited	64		(None)							
Very Droughty Loam	26	Forage	Crested Wheat Grass	1400						☐
			Intermediate Wheatgrass	1700						☑
			Western Wheatgrass	1100						☐
		Pasture	(None)							

Else if fis_method == 3 (FIS is AoA x user supplied geometry intersection boundary)

#Create FIS and associated attribute table

Copy AoA geometry to create FIS geometry

Create attribute table (one record) with following attributes

- fis_id
- AoAId
- planner_id
- inventory_date
- fis_type … value is single
- fis_area

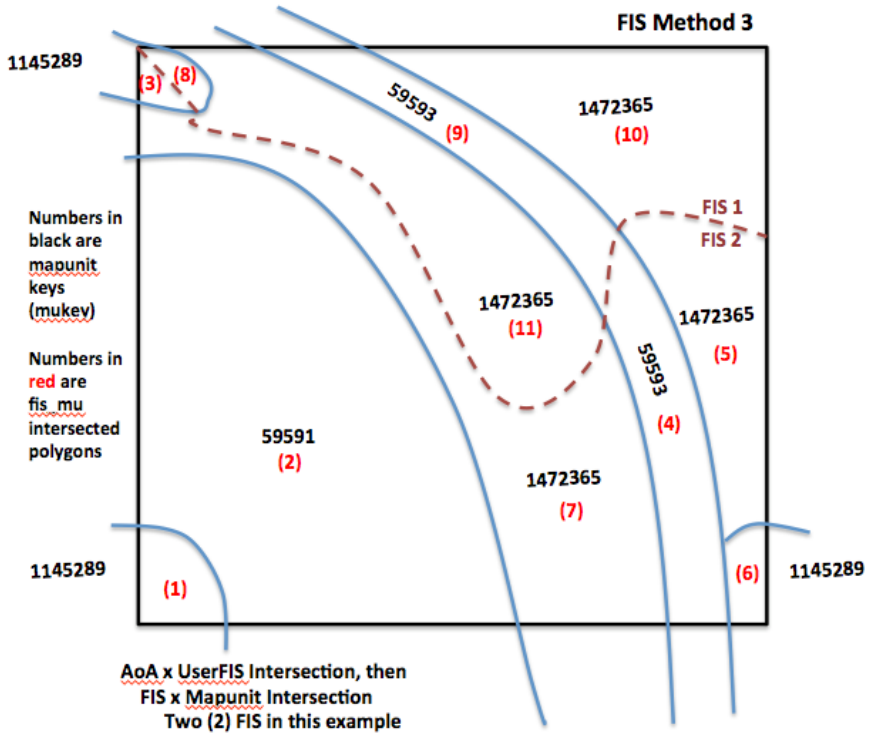
#Create FIS mapunit polygons (see following figure)

Intersect FIS geometry with SSURGO mapunit geometry

Dissolve very small intersected polygons

temp_fis_mu_attrib table columns

- fis_id … one or more per AoA
- fis_area
- fis_mu_id … one or more per FIS in the AoA
- fis_mu_area
- fis_type = 2 … multiple FIS per AoA
- AoAId
- mukey



#Create FIS mapunit component table (see following table)
Select
temp_fis_mu_attrib.AoAId
temp_fis_mu_attrib.planner_id
temp_fis_mu_attrib.inventory_date
temp_fis_mu_attrib.fis_id
temp_fis_mu_attrib.fis_area
temp_fis_mu_attrib.fis_mu_id
temp_fis_mu_attrib.fis_mu_area
component.mukey
component.cokey
component.compname
component.otherph
component.localphase
component.majcompflag
component.comppct_r
coecoclass.ecoclassid
Into temp_fis_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Inner Join coecoclass table in SSURGO
On component.cokey=coecoclass.cokey
Where coecoclass.ecoclassid LIKE 'G%' and coecoclass.ecoclasstypename = 'Forage Suitability Groups'

Order By temp_fis_mu_attrib.fis_mu_id

Alter Table temp_fis_comp
Add (fis_mu_area * comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

fis_mu_id	fis_id	mukey	cokey	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area
1	1	2581164	11328534	200	Hamar			71	142.0
		2581164	11328530		Garborg			13	26.0
		2581164	11328533		Hecla			8	16.0
		2581164	11328531		Ulen			5	10.0
		2581164	11328532		Wyndmere			3	6.0
2	1	2581147	11328413	2500	Sioux			64	1600.0
		2581147	11328415		Renshaw			26	650.0
		2581147	11328414		Fordville			4	100.0
		2581147	11328412		Arvilla			4	100.0
		2581147	11328411		Divide			2	50.0
3	1	2581164	11328534	100	Hamar			71	71.0
		2581164	11328530		Garborg			13	13.0
		2581164	11328533		Hecla			8	8.0
		2581164	11328531		Ulen			5	5.0
		2581164	11328532		Wyndmere			3	3.0
4	1	2581114	11328201	400	Cresbard			49	196.0
		2581114	11328200		Cavour			25	100.0
		2581114	11328196		Barnes			9	36.0
		2581114	11328199		Hamerly		moderately saline	5	20.0
		2581114	11328198		Ferney			5	20.0
		2581114	11328197		Svea		saline-sodic substratum	3	12.0
		2581114	11328202		Tonka			2	8.0
		2581114	11328203		Vallers		moderately saline	2	8.0
5	1	2581177	11328579	1000	Eckman			53	530.0
		2581177	11328577		Zell			20	200.0
		2581177	11328576		Gardena			13	130.0
		2581177	11328578		Barnes			4	40.0
		2581177	11328580		Glyndon			4	40.0
		2581177	11328582		Tonka			3	30.0
		2581177	11328581		Cresbard			3	30.0
6	1	2581164	11328534	50	Hamar			71	35.5
		2581164	11328530		Garborg			13	6.5
		2581164	11328533		Hecla			8	4.0
		2581164	11328531		Ulen			5	2.5
		2581164	11328532		Wyndmere			3	1.5
7	1	2581177	11328579	1500	Eckman			53	795.0
		2581177	11328577		Zell			20	300.0
		2581177	11328576		Gardena			13	195.0
		2581177	11328578		Barnes			4	60.0
		2581177	11328580		Glyndon			4	60.0
		2581177	11328582		Tonka			3	45.0
		2581177	11328581		Cresbard			3	45.0
8	2	2581114	11328201	450	Cresbard			49	220.5
		2581114	11328200		Cavour			25	112.5
		2581114	11328196		Barnes			9	40.5
		2581114	11328199		Hamerly		moderately saline	5	22.5
		2581114	11328198		Ferney			5	22.5
		2581114	11328197		Svea		saline-sodic substratum	3	13.5
		2581114	11328202		Tonka			2	9.0
		2581114	11328203		Vallers		moderately saline	2	9.0
9	2	2581177	11328579	1000	Eckman			53	530.0
		2581177	11328577		Zell			20	200.0
		2581177	11328576		Gardena			13	130.0
		2581177	11328578		Barnes			4	40.0
		2581177	11328580		Glyndon			4	40.0
		2581177	11328582		Tonka			3	30.0
		2581177	11328581		Cresbard			3	30.0
10	2	2581164	11328534	50	Hamar			71	35.5
		2581164	11328530		Garborg			13	6.5
		2581164	11328533		Hecla			8	4.0
		2581164	11328531		Ulen			5	2.5
		2581164	11328532		Wyndmere			3	1.5
11	2	2581177	11328579	500	Eckman			53	265.0
		2581177	11328577		Zell			20	100.0
		2581177	11328576		Gardena			13	65.0
		2581177	11328578		Barnes			4	20.0
		2581177	11328580		Glyndon			4	20.0
		2581177	11328582		Tonka			3	15.0
		2581177	11328581		Cresbard			3	15.0

#Create set of FIS forage suitability groups and corresponding FIS percentage
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
sum(temp_fis_comp.fis_mu_comp_pctfis) **As** fsg_pctfis
tem_fis_comp.ecoclassid
Into temp_fis_fsg table
From temp_fis_comp table
Where fsg_pctfis >= 5
Group By fis_id, ecoclassid

#Get ESIS ecological site identifiers and names for each forage suitability group
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_fsg.fis_id
temp_fis_fsg.fsg_pctfis
ecological_sites.concatenated(es_type, es_mlra, es_mlru,
es_site_number, es_state) **As** fsg_id
ecological_sites.range_site_primary_name **As** fsg_name
Into temp_fsg table
From ESIS ecological_sites table
Inner Join temp_fis_fsg table
On ecological_sites.concatenated(es_type, es_mlra, es_mlru,
es_site_number, es_state)= temp_fis_fsg.ecoclassid

#Get estimated production for each species or species mixture of the forage suitability group in the FIS
Select
temp_fsg.AoAId
temp_fsg.planner_id
temp_fsg.inventory_date
temp_fsg.fsg_id
temp_fsg.fsg_name
temp_fsg.fis_id
temp_fsg.fsg_pctfis
fsg_production.fsg_production_id
fsg_production.fsg_productiong_type
fsg_production.dry_aum_high
fsg_production.irrig_aum_high
fsg_production.dry_lbs_high

fsg_production.irrig_lbs_high
Into temp_fsg_est_prod
From ESIS fsg_production table
Inner Join temp_fsg table
On fsg_production.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_fsg.fsg_id

#Get plant symbols for each FSG production group
Select
temp_fsg est_prod.AoAId
temp_fsg est_prod.planner_id
temp_fsg est_prod.inventory_date
temp_fsg_est_prod.fsg_id
temp_fsg_est_prod.fsg_name
temp_fsg_est_prod.fis_id
temp_fsg_est_prod.fsg_pctfis
temp_fsg_est_prod.fsg_production_id
temp_fsg_est_prod.fsg_production_type
temp_fsg_est_prod.dry_aum_high
temp_fsg_est_prod.irrig_aum_high
temp_fsg_est_prod.dry_lbs_high
temp_fsg_est_prod.irrig_lbs_high
fsg_species_production_group.plant_symbol
Into temp_fsg_est_prod2
From ESIS fsg_species_production_group table
Inner Join temp_fsg_est_prod table
On fsg_species_production_group concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)= temp_fsg_est_prod.fsg_id and fsg_species_production_group.fsg_production_id= temp_fsg_est_prod.fsg_production_id

#Get plant species names for FSG plant symbols
Select
temp_fsg_est_prod2.fsg_id
temp_fsg_est_prod2.fis_id
temp_fsg_est_prod2.fsg_production_id
plant_master.vernacular
Into temp_fsg_species
From plant_master
Inner Join temp_fsg_est_prod2 **On** plant_master.symbol= temp_fsg_est_prod2.plant_symbol

#Following probably is not correct, but plant species names for a fsg production group must be concantenated into a single data field
Select
temp_fsg_species.fsg_id
temp_fsg_species.fis_id

```
    temp_fsg_species.fsg_production_id
    Stuff((
        Select ',' + temp_fsg_species.vernacular As [text()]
        #Add a comma (,) before each value
        From temp_fsg_species
        #Select it as XML and remove 1st character (,) from result
        For XML PATH('') ), 1, 1, '' )
        As temp_fsg_species_list
Into temp_fsg_species2
From temp_fsg_species



#Complete FSG estimated production table
Select
    temp_fsg est_prod2.AoAId
    temp_fsg est_prod2.planner_id
    temp_fsg est_prod2.inventory_date
    temp_fsg_est_prod2.fsg_id
    temp_fsg_est_prod2.fsg_name
    temp_fsg_est_prod2.fis_id
    temp_fsg_est_prod2.es_pctfis
    temp_fsg_est_prod2.fsg_production_id
    temp_fsg_species2.fsg_species_list
    temp_fsg_est_prod2.fsg_production_type
    temp_fsg_est_prod2.dry_aum_high
    temp_fsg_est_prod2.irrig_aum_high
    temp_fsg_est_prod2.dry_lbs_high
    temp_fsg_est_prod2.irrig_lbs_high
Into temp_fsg_est_prod3
From temp_fsg_est_prod2
Inner Join temp_fsg_species2
On temp_fsg_est_prod2.fsg_id= temp_fsg_species2.fsg_id and
temp_fsg_est_prod2.fis_id= temp_fsg_species2.fis_id and
temp_fsg_est_prod2.fsg_production_id=
temp_fsg_species2.fsg_production_id
Group By fsg_species_list, fsg_production_id

#Send to Output
Output contents of temp_fsg_est_prod3 table

#Create list of soil components for each FSG in the FIS
Select
    temp_fis_comp.fis_id
    temp_fis_comp.cokey
    temp_fis_comp.compname
    temp_fis_comp.ecoclassid As fsg_id
```

Into temp_fsg_comp
From temp_fis_comp
Group By ecoclassid, compname

#Get maximum irrigated and non-irrigated forage production from SSURGO for each soil component crop represented in the FIS
Select
temp_fsg_comp.fsg_id
temp_fsg_comp.fis_id
temp_fsg_comp.cokey
cocropyld.cocropyldkey
cocropyld.cropname
max(cocropyld.nonirryield_r) **As** nonirryld_aum
max(cocropyld.irryield_r) **As** irryld_aum
Into temp_fsg_comp_est_prod
From SSURGO cocropyld table
Inner Join temp_fsg_comp table **On** cocropyld.cokey= temp_fsg_comp.cokey
Where cocropyld.yldunits=AUM
Group By fsg_id, cropname

#Send to Output
Output data in temp_fsg_comp_est_production

#Create list of soil mapunits for each FSG in the FIS
Select
temp_fis_comp.fis_id
temp_fis_comp.mukey
temp_fis_comp.muname
temp_fis_comp.ecoclassid **As** fsg_id
Into temp_fsg_comp
From temp_fis_comp
Group By ecoclassid, muname

#Get maximum irrigated and non-irrigated forage production from SSURGO for each soil mapunit crop represented in the FIS
Select
temp_fsg_comp.fsg_id
temp_fsg_comp.fis_id
temp_fsg_comp.mukey
mucropyld.mucropyldkey
mucropyld.cropname
max(mucropyld.nonirryield_r) **As** nonirryld_aum
max(mucropyld.irryield_r) **As** irryld_aum
Into temp_fsg_mu_est_prod
From SSURGO mucropyld table
Inner Join temp_fsg_comp table **On** mucropyld.cokey= temp_fsg_comp.cokey

Where mucropyld.yldunits=AUM
Group By fsg_id, cropname

#Send to Output
Output data in temp_fsg_mu_est_production

#The data in es_est_prod should enable the application to create the following choice list

FIS 1		FIS 2								
Forage Suitability Group	% of FIS	Producton Type	Plant Species	FSG Dry (lbs/ac)	FSG Dry (lbs/ac)	FSG Dry (AUM)	FSG Irrig (AUM)	SSURGO Dry (AUM)	SSURGO Irrig (AUM)	Select
fsg_name	calc									
Loam	42	Forage	Alfalfa	3900						☑
			Alfalfa, Intermediate Wheatgrass	3500						☐
			Smooth Brome, Alfalfa	3500						☐
			Big Bluestem	2900						☐
			Crested Wheat Grass	2600						☐
			Green Needlegrass	2000						☐
			Intermediate Wheatgrass	2800						☐
			Smooth Brome	2800						☐
			Switchgrass	3100						☐
			Western Wheatgrass	2000						☐
		Pasture	(None)							☐
Limy Upland	15	Forage	Crested Wheat Grass, Alfalfa	2400						☐
			Alfalfa, Intermediate Wheatgrass	2700						☐
			Crested Wheat Grass	2400						☐
			Intermediate Wheatgrass	2500						☐
			Little Bluestem	2300						☐
		Pasture	(None)							☐
Clayey Subsoil	11	Forage	Alfalfa	3400						☐
			Alfalfa, Intermediate Wheatgrass	3200						☐
			Smooth Brome, Alfalfa	3200						☐
			Big Bluestem	2600						☐
			Crested Wheat Grass	2500						☐
			Green Needlegrass	1600						☐
			Intermediate Wheatgrass	2300						☐
			Smooth Brome	2300						☐
			Switchgrass	2900						☐
			Western Wheatgrass	1900						☐
		Pasture	(None)							☐
Overflow	10	Forage	Alfalfa	5400						☐
			Alfalfa, Intermediate Wheatgrass	4000						☐
			Smooth Brome, Alfalfa	4000						☐
			Big Bluestem	3700						☐
			Indiangrass	3100						☐
			Intermediate Wheatgrass	3100						☐
			Smooth Brome	3100						☐
			Switchgrass	4000						☐
		Pasture	(None)							☐
Claypan	6	Forage	Alfalfa	2600						☐
			Crested Wheat Grass	2100						☐
			Intermediate Wheatgrass	2300						☐
			Crested Wheat Grass	1800						☐
			Intermediate Wheatgrass	1800						☐
			Tall Wheatgrass	1800						☐
			Western Wheatgrass	1300						☐
		Pasture	(None)	NA						☐

1.2. Output
AoAId … one
planner_id

Inventory_date
fis_type
FIS identifier (fis_id) … one or more per AoA depending on FIS method
    FIS polygon geometry
    fsg_id … one or more forage suitability groups per FIS
        fsg_name
        fsg_pctfis
        fsg_production_id … one or more per FSG
            fsg_species_list
            fsg_production_type … one or two (forage, pasture)
                dry_aum_high
                irrig_aum_high
                dry_lbs_high
                irrig_lbs_high
        cocropyldkey … one or more per FSG
            cropname
            nonirryield_aum
            irryield_aum
        mucropyldkey … one or more per FSG
            cropname
            nonirryield_aum
            irryield_aum

Service GRAS-1c: Create Forage Inventory Sites, Get Soil Component/Mapunit Estimated Production for an Area of Analysis (FISProdSSURGO)

Purpose: Create forage inventory sites (FISs) for an area of analysis (AoA) and get estimated production from SSURGO soil components or mapunits.

This service is similar to GRAS-1a and 1b except it supports get estimated production method 3 – From SSURGO. SSURGO contains three sources of estimated production: (1) soil component range forage, (2) soil component irrigated and non-irrigated harvested forage by crop or species mix, and (3) soil mapnunit irrigated and non-irrigated forage by crop or species mix.

The six methods (abstracted to three) in GRAS-1a and 1b for creating FIS polygons apply to this service. Any requires the creation of a table containing soil components relevant to the FIS. SSURGO components can contain dry matter production values or link to mapunits containing dry matter production values, the latter (if available) serving as backup when soil components do not have species groups and associated forage production.

This service consumes an application request payload to create one or more FIS per AoA, their soil compoents (above 5% of FIS area), and get estimated dry matter production for associated plants/crops. The results payload returns data enabling the requesting application to populate a choice list for associating estimated dry matter production to a FIS or FIS component.

Service Signature

Request Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
AoA polygon geometry … one set of coordinates per AoA, Area of Analysis Geometry
aoa_land_use … integer, corresponding to NRCS land_use_id; choices allowed are 1 – crop, 2 – forest, 3 – range, 4 – pasture, 5 – Protected, 9 – Other Rural Land, and 10 – Associated Agricultural Land
fis_method … integer, choices are 1 - FIS is AoA boundary, 2 – FIS is AoA x mapunit intersection boundary, 3 – FIS is AoA x user supplied geometry intersection boundary
est_prod_method … integer, value must be 3 – from SSURGO component or mapunit
User FIS geometry … coordinate set for intersecting with AoA geometry (GPS, digitized, copied polygons), only if FIS method 3
planner_id … char varying(23)… application user identifier
inventory_date … date (yyyy-mm-dd), Forage Inventory Date

Result Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
planner_id … char varying(23)… application user identifier
inventory_date … date (yyyy-mm-dd), Forage Inventory Date
fis_type … character varying, Forage Inventory Site Type
fis_id … integer, one or more per AoA depending on FIS method, Forage Inventory Site Identifier

FIS polygon geometry, one set of coordinates for the FIS, Forage Inventory Site Geometry
cokey … character varying(60), one or more in the FIS, Soil Component Key
compname … character varying(120), Soil Component Name
otherph … character varying, Soil Component Other Phase Name
localphase … character varying, Soil Component Local Phase Name
majcompflag … character varying(6), Major Component Flag (Yes/No)
fis_mu_comp_agg_pct … numeric(3,0), Aggregated percentage of components for similar soil map units within FIS
ecoclassid … character varying(60), Ecological Site Identifier
ecoclassname … text, Ecological Site Name
rsprod_r … bigint … Representative Ecological Site Forage Production
cocropyldkey … character varying(60), one or more for the soil component (cokey), Soil Component Crop Yield Key
cropname … character varying(508), Crop Name
nonirryld_aum … numeric, Non-Irrigated Yield in AUMs
irryld_aum … numeric, Irrigated Yield in AUMs
mukey … character varying(6), Soil Mapunit Key
mucropyldkey … character varying(60), one or more for the soil component (cokey), Mapunit Crop Yield Key
cropname … character varying(508), Crop Name
nonirryld_aum … numeric, Non-Irrigated Yield in AUMs
irryld_aum … numeric, Irrigated Yield in AUMs

Reference Data Sources

SSURGO Data Mart
mapunit table
component table
coecoclass table
cocropyld table
mucropyld table

Component

1. **Create Forage Inventory Sites (FISdelineate)**
   1.1. Inputs

AoAId	1	2	3
aoa_geometry	[40.6735513978082, -103.809573763755], [40.6607911607823, -103.82648240938], [40.6735513978082, -103.82648240938], [40.6607911607823, -103.809573763755]	[40.6689809721788, -103.8472001256], [40.6434563076883, -103.896295279396], [40.6689809721788, -103.896295279396], [40.6434563076883, -103.8472001256]	[40.6989209302277, -103.972512930964], [40.6502291091709, -104.086496085249], [40.6989209302277, -104.086496085249], [40.6502291091709, -103.972512930964]
aoa_land_use	3	1	4
fis_method	1	2	3
user_fis_geometry			[40.710307599928, -104.085122794275], [40.6295829439063, -104.002038687843]
planner_id	Dana Smith	Dana Smith	Jan Jones
inventory_date	20141003	20140915	20150223

1.2. GIS Operations

For each AoA

#Grazing unit (AoA) is the the Forage Inventory Site!!

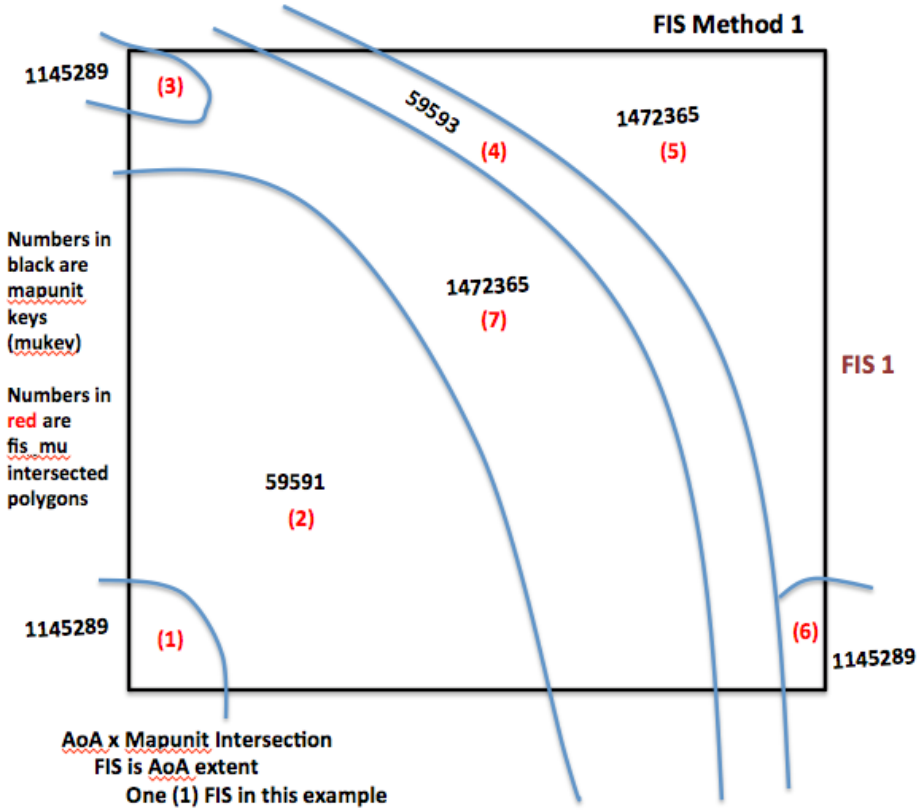
If fis_method == 1 (FIS congruent with AoA boundary)

#Create FIS and associated attribute table

Copy AoA geometry to create FIS geometry

Create attribute table (one record) with following attributes

- fis_id
- AoAId
- planner_id
- inventory_date
- fis_type … value is single
- fis_area



#Create FIS mapunit polygons (see figure above)
Intersect FIS geometry with SSURGO mapunit geometry
Dissolve very small intersected polygons

temp_fis_mu_attrib table
- fis_id … one per AoA
- fis_area
- fis_type = 1 … single FIS per AoA
- AoAId
- planner_id
- inventory_date
- fis_mu_id … one or more per AoA
- mukey
- fis_mu_area

#Create FIS mapunit component table (see following table)
Select
- temp_fis_mu_attrib.AoAId
- temp_fis_mu_attrib.planner_id
- temp_fis_mu_attrib.inventory_date
- temp_fis_mu_attrib.fis_id
- temp_fis_mu_attrib.fis_area
- temp_fis_mu_attrib.fis_mu_id
- temp_fis_mu_attrib.fis_mu_area

component.mukey
component.cokey
component.compname
component.otherph
component.localphase
component.majcompflag
component.comppct_r
Into temp_fis_mu_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Order By fis_mu_id

Alter Table temp_fis_mu_comp
Add (fis_mu_area * comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

#Note: not all of the data elements selected for fis_mu_comp displayed in the following graphic

fis_mu_id	fis_id	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area	fis_mu_comp_pctfis
1	1	200	Carwile			65	130	1.7%
			Eda			30	60	0.8%
			Grandfield			5	10	0.1%
2	1	2500	Devol			5	132	1.7%
			Nobscot			80	2105	27.2%
			Delwin		moist	4	105	1.4%
			Carwile			1	26	0.3%
			Grandfield			5	132	1.7%
3	1	100	Carwile			65	65	0.8%
			Eda			30	30	0.4%
			Grandfield			5	5	0.1%
4	1	850	Oklark			5	43	0.5%
			Abbie			85	723	9.3%
			St. Paul			7	60	0.8%
			Otero			3	26	0.3%
5	1	2000	Obaro			25	588	7.6%
			Quinlan			60	1412	18.2%
6	1	100	Carwile			65	65	0.8%
			Eda			30	30	0.4%
			Grandfield			5	5	0.1%
7	1	2000	Obaro			25	588	7.6%
			Quinlan			60	1412	18.2%

#Create list of soil components for the AoA-extent FIS
Select
temp_fis_mu_comp.AoAId
temp_fis_mu_comp.planner_id
temp_fis_mu_comp.inventory_date
temp_fis_mu_comp.fis_id
temp_fis_mu_comp.cokey
temp_fis_mu_comp.compname
temp_fis_mu_comp.otherph

temp_fis_mu_comp.localphase
temp_fis_mu_comp.majcompflag
temp_fis_mu_comp.mukey
SUM temp_fis_mu_comp. fis_mu_comp_area **As** fis_mu_comp_agg_area
SUM temp_fis_mu_comp. fis_mu_comp_pctfis **As** fis_mu_comp_agg_pct
Into temp_fis_comp
From temp_fis_mu_comp
Group By cokey, compname, otherph, localphase, majcompflag, mukey
#Two or more mapunits may have the same component, but the components in common often will have different cokey values; grouping should combine records having the same cokey. The reason for a component having different different cokey values is that soil scientists have not finished correlation of components across mapunits.

#For range and other not cultivated land use (excluding forest), get representative range production for soil components in AoA-extent FIS
If aoa_land_use == 3, 5, 9, or 10
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
component.rsprod_r
coecoclass.ecoclassid
coecoclass.ecoclassname
Into temp_fis_comp_rsprod
From component table in SSURGO
Inner Join temp_fis_comp
Inner Join coecoclass
On component.cokey= temp_fis_comp.cokey
On component.cokey=coecoclass.cokey
Where coecoclassid **LIKE** 'R%' and fis_mu_comp_agg_pct >= 5

#For forest land use, get representative estimated production for soil components in AoA extent FIS
Else if aoa_land_use ==2
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey

temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
component.rsprod_r
coecoclass.ecoclassid
coecoclass.ecoclassname
Into temp_fis_comp_rsprod
From component table in SSURGO
Inner Join temp_fis_comp
Inner Join coecoclass
On component.cokey= temp_fis_comp.cokey
On component.cokey=coecoclass.cokey
Where coecoclassid **LIKE** 'F%' and fis_mu_comp_agg_pct >= 5

#Send to Output
Output data from temp_fis_comp_rsprod

#The data in fis_comp_rsprod should enable the application to create the following choice list

FIS 1 FIS Number	Component Name	Other Phase	Local Phase	% of FIS	Ecological Site Name	SSURGO Range Production (lbs/ac)	Select
1	Quinlan			37	Shallow 19-26" PZ	1800	☐
	Nobscot			27	Sandy 19-26" PZ	2800	☐
	Obaro			15	Loamy Prairie 19-26" PZ	1800	☐

#For cultivated land use, get maximum irrigated and non-irrigated forage production from SSURGO for forage crops associated with the soil components of the AoA-extent FIS
Else if aoa_land_use == 1 or 4
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.compname
temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
cocropyld.cropname
max(cocropyld.nonirryield_r) **As** nonirryld_aum
max(cocropyld.irryield_r) **As** irryld_aum
Into temp_fis_comp_est_prod
From SSURGO cocropyld table
Inner Join temp_fis_comp table **On** cocropyld.cokey=

temp_fis_comp.cokey
Where cocropyld.yldunits=AUM and fis_mu_comp_agg_pct >= 5

#Send to Output
Output data from temp_fis_comp_est_prod table

#Get maximum irrigated and non-irrigated forage production from SSURGO for each soil mapunit crop represented in the FIS
Select
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.mukey
mucropyld.mucropyldkey
mucropyld.cropname
max(mucropyld.nonirryield_r) **As** nonirryld_aum
max(mucropyld.irryield_r) **As** irryld_aum
Into temp_fis_mu_est_prod
From SSURGO mucropyld table
Inner Join temp_fis_comp
On mucropyld.mukey= temp_fis_comp.mukey
Where mucropyld.yldunits=AUM

#Send to Output
Output from temp_fis_mu_est_prod table

#The data in fis_comp_est_prod and fis_mu_est_prod should enable the application to create the following choice list

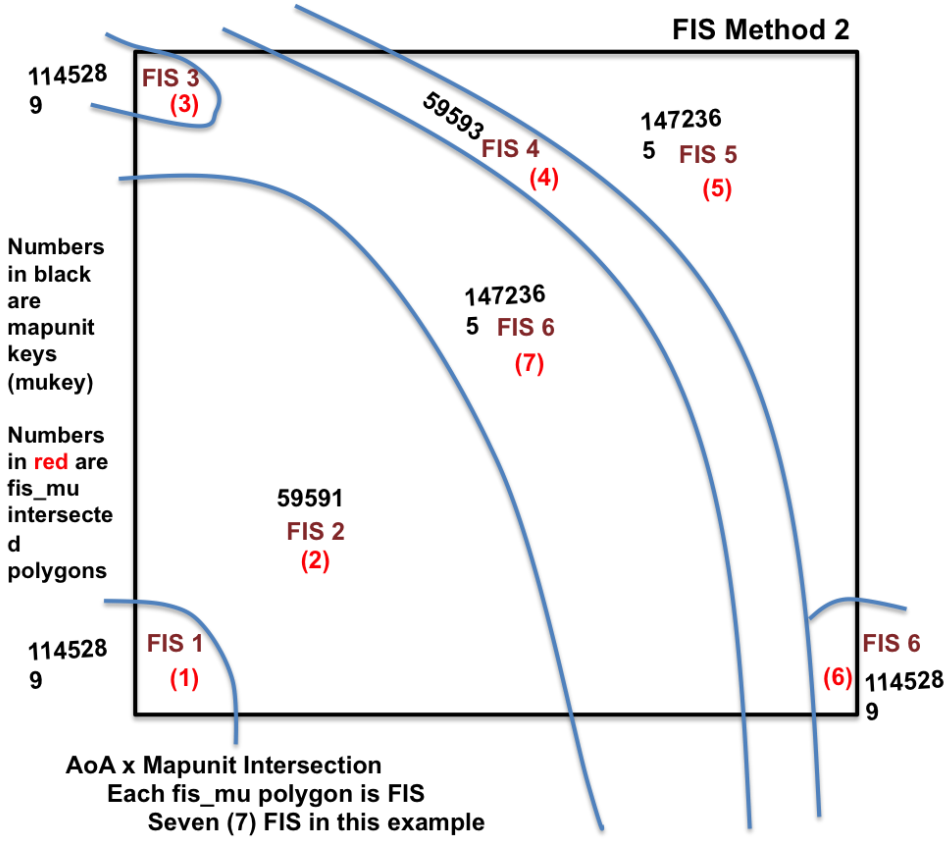
FIS 1 FIS Number	Component Name	Other Phase	Local Phase	% of FIS	Production Type	Crop Name	SSURGO Dry (AUM)	SSURGO Irrig (AUM)	Select
1	Quinlan			37	Component	Improved bermudagrass	1.5		☐
						Introduced bluestem	2.3		☑
						Weeping lovegrass	1.5		☐
	Nobscot			27	Component	Sorghum grazed	3.1		☐
						Improved bermudagrass	4		☐
						Weeping lovegrass	5		☐
						Small grains grazeout	1.9		☐
	Obaro			15	Component	Improved bermudagrass	2.5		☐
						Introduced bluestem	2.4		☐
						Weeping lovegrass	3		☐
	Abbie			9	Mapunit	Introduced bluestem	5.8		☐
						Improved bermudagrass	5	9	☐

#AoA x Mapunit intersected polygon is the FIS!!
Else if fis_method == 2 (FIS = fis_mu_id)
#Create one or more FIS within the AoA
Intersect AoA and SSURGO mapunits to create one or more FIS polygons
Dissolve very small intersected polygons in the AoA

#Establish FIS mapunit attribute table

temp_fis_mu_attrib table columns

- fis_id … one or more in the AoA
- fis_area
- AoAId … one per fis_id
- planner_id … one per fis_id
- inventory_date … one per fis_id
- fis_type = 2 … multiple FIS per AoA
- mukey … one associated with fis_id



#Create FIS mapunit component table (see following table)

Select

- temp_fis_mu_attrib.AoAId
- temp_fis_mu_attrib.planner_id
- temp_fis_mu_attrib.inventory_date
- temp_fis_mu_attrib.fis_id
- temp_fis_mu_attrib.fis_area
- component.mukey
- component.cokey
- component.compname
- component.otherph
- component.localphase
- component.majcompflag

component.comppct_r
Into temp_fis_mu_comp table
From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Order By temp_fis_mu_attrib.fis_id

Alter Table temp_fis_mu_comp
Add (fis_area* comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

fis_mu_id	fis_id	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area	fis_mu_comp_pctfis
1	1	200	Carwile			65	130	65%
			Eda			30	60	30%
			Grandfield			5	10	5%
2	2	2500	Devol			5	132	5%
			Nobscot			80	2105	84%
			Delwin		moist	4	105	4%
			Carwile			1	26	1%
			Grandfield			5	132	5%
3	3	100	Carwile			65	65	65%
			Eda			30	30	30%
			Grandfield			5	5	5%
4	4	850	Oklark			5	43	5%
			Abbie			85	723	85%
			St. Paul			7	60	7%
			Otero			3	26	3%
5	5	2000	Obaro			25	588	29%
			Quinlan			60	1412	71%
6	6	100	Carwile			65	65	65%
			Eda			30	30	30%
			Grandfield			5	5	5%
7	7	2000	Obaro			25	588	29%
			Quinlan			60	1412	71%

#Create list of soil components for each mapunit-based FIS
Select
temp_fis_mu_comp.AoAId
temp_fis_mu_comp.planner_id
temp_fis_mu_comp.inventory_date
temp_fis_mu_comp.fis_id
temp_fis_mu_comp.cokey
temp_fis_mu_comp.compname
temp_fis_mu_comp.otherph
temp_fis_mu_comp.localphase
temp_fis_mu_comp.majcompflag
temp_fis_mu_comp.mukey
SUM temp_fis_mu_comp. fis_mu_comp_area **As** fis_mu_comp_agg_area
SUM temp_fis_mu_comp. fis_mu_comp_pctfis **As** fis_mu_comp_agg_pct
Into temp__comp
From temp_fis_mu_comp
Group By cokey, compname, otherph, localphase, majcompflag, mukey

#Two or more mapunits may have the same component, but the components in common often will have different cokey values; grouping should combine records having the same cokey. The reason for a component having different different cokey values is that soil scientists have not finished correlation of components across mapunits.

#For range and other not cultivated land use (excluding forest), get representative range production for soil components in AoA-extent FIS

If aoa_land_use == 3, 5, 9, or 10

Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
component.rsprod_r
coecoclass.ecoclassid
coecoclass.ecoclassname
Into temp_fis_comp_rsprod
From component table in SSURGO
Inner Join temp_fis_comp
Inner Join coecoclass
On component.cokey= temp_fis_comp.cokey
On component.cokey=coecoclass.cokey
Where coecoclassid **LIKE** 'R%' and fis_mu_comp_agg_pct >= 5

#For forest land use, get representative estimated production for soil components in AoA extent FIS

Else if aoa_land_use ==2

Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
component.rsprod_r
coecoclass.ecoclassid
coecoclass.ecoclassname
Into temp_fis_comp_rsprod

From component table in SSURGO
Inner Join temp_fis_comp
Inner Join coecoclass
On component.cokey= temp_fis_comp.cokey
On component.cokey=coecoclass.cokey
Where coecoclassid **LIKE** 'F%' and fis_mu_comp_agg_pct >= 5

#Send to Output
Output data from temp_fis_comp_rsprod

#The data in fis_comp_rsprod should enable the application to create the following choice list

FIS 1	FIS 2	FIS 3	FIS 4	FIS 5	FIS 6	FIS 7		
FIS Number	Component Name	Other Phase	Local Phase	% of FIS	Ecological Site Name		SSURGO Range Production (lbs/ac)	Select
2	Nobscot			80	Sandy 19-26" PZ		2800	☐
	Devol			5	Loamy Sand Prairie 19-26" PZ		2600	☐
	Grandfield			5	Sandy Loam 19-26" PZ		3200	☐

#For cultivated land use, get maximum irrigated and non-irrigated forage production from SSURGO for forage crops associated with the soil components of the AoA-extent FIS
Else if aoa_land_use == 1 or 4
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.compname
temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
cocropyld.cropname
max(cocropyld.nonirryield_r) **As** nonirryld_aum
max(cocropyld.irryield_r) **As** irryld_aum
Into temp_fis_comp_est_prod
From SSURGO cocropyld table
Inner Join temp_fis_comp table **On** cocropyld.cokey=
temp_fis_comp.cokey
Where cocropyld.yldunits=AUM and fis_mu_comp_agg_pct >= 5

#Send to Output
Output data from temp_fis_comp_est_prod table

#Get maximum irrigated and non-irrigated forage production from SSURGO for each soil mapunit crop represented in the FIS

Select
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.mukey
mucropyld.mucropyldkey
mucropyld.cropname
max(mucropyld.nonirryield_r) **As** nonirryld_aum
max(mucropyld.irryield_r) **As** irryld_aum
Into temp_fis_mu_est_prod
From SSURGO mucropyld table
Inner Join temp_fis_comp
On mucropyld.mukey= temp_fis_comp.mukey
Where mucropyld.yldunits=AUM

#Send to Output
Output from temp_fis_mu_est_prod table

#The data in es_est_prod should enable the application to create the following choice list

FIS 1	FIS 2	FIS 3	FIS 4	FIS 5	FIS 6	FIS 7			
FIS Number	Component Name	Other Phase	Local Phase	% of FIS	Production Type	Crop Name	SSURGO Dry (AUM)	SSURGO Irrig (AUM)	Select
2	Nobscot			80	Component	Sorghum grazed	3.1		☐
						Improved bermudagrass	4		☑
						Weeping lovegrass	5		☐
						Small grains grazeout	1.9		☐
	Devol			5	Component	Improved bermudagrass	5		☐
						Weeping lovegrass	5.5		☐
	Grandfield			5	Component	Improved bermudagrass	5		☐
						Weeping lovegrass	6		☐

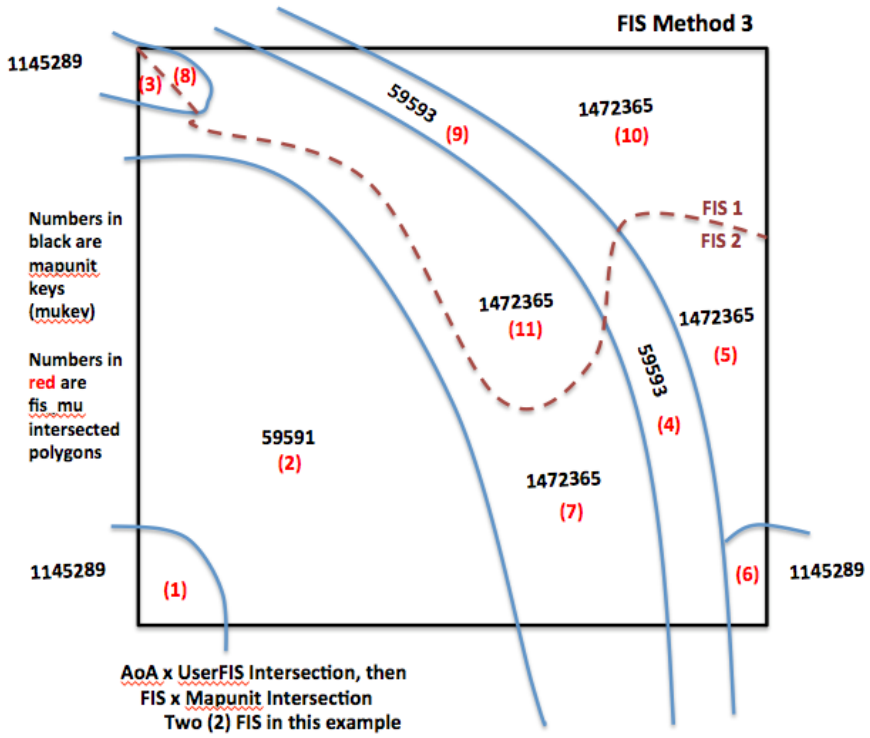
Else if fis_method == 3 (FIS is AoA x user supplied geometry intersection boundary)

#Create FIS and associated attribute table
Copy AoA geometry to create FIS geometry
Create attribute table (one record) with following attributes
fis_id
AoAId
planner_id
inventory_date
fis_type … value is single
fis_area

#Create FIS mapunit polygons (see following figure)
Intersect FIS geometry with SSURGO mapunit geometry
Dissolve very small intersected polygons

temp_fis_mu_attrib table columns
fis_id … one or more per AoA
fis_area

fis_mu_id … one or more per FIS in the AoA
fis_mu_area
fis_type = 2 … multiple FIS per AoA
AoAId
planner_id
inventory_date
mukey



#Create FIS mapunit component table (see following table)
Select
temp_fis_mu_attrib.AoAId
temp_fis_mu_attrib.planner_id
temp_fis_mu_attrib.inventory_date
temp_fis_mu_attrib.fis_id
temp_fis_mu_attrib.fis_area
temp_fis_mu_attrib.fis_mu_id
temp_fis_mu_attrib.fis_mu_area
component.mukey
component.cokey
component.compname
component.otherph
component.localphase
component.majcompflag
component.comppct_r
Into temp_fis_mu_comp table

From temp_fis_mu_attrib table
Inner Join component table in SSURGO
On temp_fis_mu_attrib.mukey=component.mukey
Order By temp_fis_mu_attrib.fis_mu_id

Alter Table temp_fis_mu_comp
Add (fis_mu_area * comppct_r) **As** fis_mu_comp_area
Add (fis_mu_area * comppct_r / fis_area) **As** fis_mu_comp_pctfis

fis_mu_id	fis_id	fis_mu_area	compname	otherph	localphase	comppct_r	fis_mu_comp_area	fis_mu_comp_pctfis
1	1	200	Carwile			65	130	3%
			Eda			30	60	1%
			Grandfield			5	10	0%
2	1	2500	Devol			5	131.6	3%
			Nobscot			80	2105.3	44%
			Delwin		moist	4	105.3	2%
			Carwile			1	26.3	1%
			Grandfield			5	131.6	3%
3	1	100	Carwile			65	65	1%
			Eda			30	30	1%
			Grandfield			5	5	0%
4	1	400	Oklark			5	20.0	0%
			Abbie			85	340.0	7%
			St. Paul			7	28.0	1%
			Otero			3	12.0	0%
5	1	1000	Obaro			25	250.0	5%
			Quinlan			60	600.0	13%
6	1	50	Carwile			65	32.5	1%
			Eda			30	15.0	0%
			Grandfield			5	2.5	0%
7	1	1000	Obaro			25	250.0	5%
			Quinlan			60	600.0	13%
8	2	450	Oklark			5	22.5	1%
			Abbie			85	382.5	15%
			St. Paul			7	31.5	1%
			Otero			3	13.5	1%
9	2	1000	Obaro			25	250.0	10%
			Quinlan			60	600.0	24%
10	2	50	Carwile			65	32.5	1%
			Eda			30	15.0	1%
			Grandfield			5	2.5	0%
11	2	1000	Obaro			25	250.0	10%
			Quinlan			60	600.0	24%

#Create list of soil components for each FIS in the AoA
Select
temp_fis_mu_comp.AoAId
temp_fis_mu_comp.planner_id
temp_fis_mu_comp.inventory_date
temp_fis_mu_comp.fis_id
temp_fis_mu_comp.cokey
temp_fis_mu_comp.compname
temp_fis_mu_comp.otherph

temp_fis_mu_comp.localphase
temp_fis_mu_comp.majcompflag
temp_fis_mu_comp.mukey
SUM temp_fis_mu_comp. fis_mu_comp_area **As** fis_mu_comp_agg_area
SUM temp_fis_mu_comp. fis_mu_comp_pctfis **As** fis_mu_comp_agg_pct
Into temp_fis_comp
From temp_fis_mu_comp
Group By cokey, compname, otherph, localphase, majcompflag, mukey
#Two or more mapunits may have the same component, but the components in common often will have different cokey values; grouping should combine records having the same cokey. The reason for a component having different different cokey values is that the soil survey program has not finished correlation of components across mapunits.

#For range and other not cultivated land use (excluding forest), get representative range production for soil components in AoA-extent FIS
If aoa_land_use == 3, 5, 9, or 10
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
component.rsprod_r
coecoclass.ecoclassid
coecoclass.ecoclassname
Into temp_fis_comp_rsprod
From component table in SSURGO
Inner Join temp_fis_comp
Inner Join coecoclass
On component.cokey= temp_fis_comp.cokey
On component.cokey=coecoclass.cokey
Where coecoclassid **LIKE** 'R%' and fis_mu_comp_agg_pct >= 5

#For forest land use, get representative estimated production for soil components in AoA extent FIS
Else if aoa_land_use ==2
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey

temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
component.rsprod_r
coecoclass.ecoclassid
coecoclass.ecoclassname
Into temp_fis_comp_rsprod
From component table in SSURGO
Inner Join temp_fis_comp
Inner Join coecoclass
On component.cokey= temp_fis_comp.cokey
On component.cokey=coecoclass.cokey
Where coecoclassid **LIKE** 'F%' and fis_mu_comp_agg_pct >= 5

#Send to Output
Output data from temp_fis_comp_rsprod

#The data in fis_comp_rsprod should enable the application to create the following choice list

FIS 1	FIS 2						
FIS Number	Component Name	Other Phase	Local Phase	% of FIS	Ecological Site Name	SSURGO Range Production (lbs/ac)	Select
2	Quinlan			49	Shallow 19-26" PZ	1800	☐
	Obaro			20	Loamy Prairie 19-26" PZ	1800	☐
	Abbie			16	(None)	(None)	

#For cultivated land use, get maximum irrigated and non-irrigated forage production from SSURGO for forage crops associated with the soil components of the AoA-extent FIS
Else if aoa_land_use == 1 or 4
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.compname
temp_fis_comp.otherph
temp_fis_comp.localphase
temp_fis_comp.majcompflag
temp_fis_comp.fis_mu_comp_agg_pct
cocropyld.cropname
max(cocropyld.nonirryield_r) **As** nonirryld_aum
max(cocropyld.irryield_r) **As** irryld_aum
Into temp_fis_comp_est_prod
From SSURGO cocropyld table
Inner Join temp_fis_comp table **On** cocropyld.cokey=

temp_fis_comp.cokey
Where cocropyld.yldunits=AUM and fis_mu_comp_agg_pct >= 5

#Send to Output
Output data from temp_fis_comp_est_prod table

#Get maximum irrigated and non-irrigated forage production from SSURGO for each soil mapunit crop represented in the FIS
Select
temp_fis_comp.AoAId
temp_fis_comp.planner_id
temp_fis_comp.inventory_date
temp_fis_comp.fis_id
temp_fis_comp.cokey
temp_fis_comp.mukey
mucropyld.mucropyldkey
mucropyld.cropname
max(mucropyld.nonirryield_r) **As** nonirryld_aum
max(mucropyld.irryield_r) **As** irryld_aum
Into temp_fis_mu_est_prod
From SSURGO mucropyld table
Inner Join temp_fis_comp
On mucropyld.mukey= temp_fis_comp.mukey
Where mucropyld.yldunits=AUM

#Send to Output
Output from temp_fis_mu_est_prod table

#The data in es_est_prod should enable the application to create the following choice list

FIS 1	FIS 2								
FIS Number	Component Name	Other Phase	Local Phase	% of FIS	Production Type	Crop Name	SSURGO Dry (AUM)	SSURGO Irrig (AUM)	Select
2	Quinlan			49	Component	Improved bermudagrass	1.5		☐
						Introduced bluestem	2.3		☑
						Weeping lovegrass	1.5		☐
	Obaro			20	Component	Improved bermudagrass	2.5		☐
						Introduced bluestem	2.4		☐
						Weeping lovegrass	3		☐
	Abbie			16	Mapunit	Introduced bluestem	5.8		☐
						Improved bermudagrass	5	9	☐

1.3. Output
AoAId … one
planner_id
Inventory_date
fis_type
fis_id … one or more in the AoA
FIS polygon geometry
cokey … one or more in the FIS (fis_id)

compname
otherph
localphase
majcompflag
fis_mu_comp_agg_pct
ecoclassid
ecoclassname
rsprod_r
cocropyldkey
    cropname
    nonirryld_aum
    irryld_aum
mukey
mucropyldkey … one or more for the soil component (cokey)
    cropname
    nonirryld_aum
    irryld_aum

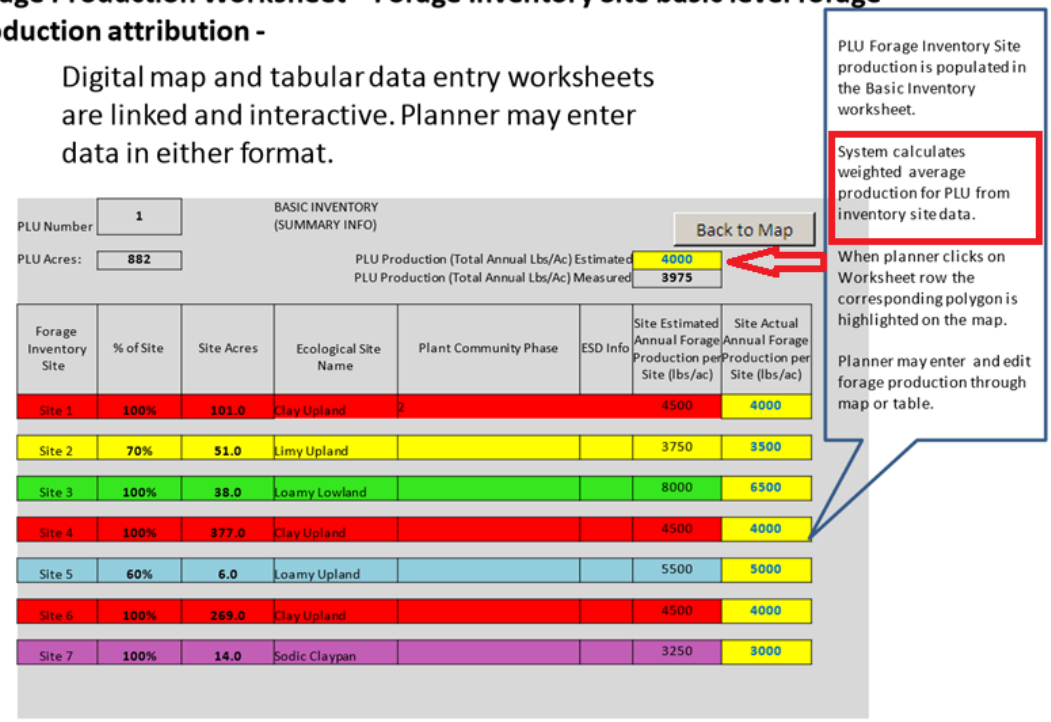
Service GRAS-1d: Calculate Weighted Average Estimated and Measured Forage Production Values (CalcWtAvProd)

Purpose: Calculate weighted average estimated and measured forage production values for the AoA and forage inventory sites.

This service calculates a weighted average estimated and measured forage production value for each forage inventory site. The FIS weighted average estimated and measured forage production values are then utilized to calculate a weighted average estimated and measured forage production values for the AoA.

Forage Production Worksheet– Forage Inventory Site basic level forage production attribution -

Digital map and tabular data entry worksheets are linked and interactive. Planner may enter data in either format.



Service Signature

Request Payload

AoAId … integer, one in the request; Area of Analysis Identifier
aoa_acres … decimal(10,2); Area of Analysis Acres
fis_id … integer, one or more per AoA; Forage Inventory Site Identifier
fis_acres … decimal(10, 2); Forage Inventory Site Acres

fis_component_id … varchar(50), one, two, or three per FIS; Forage Inventory Site Component Identifier
forage_production_estimated … decimal(10, 2); Estimated Forage Production in Pounds Per Acre

forage_production_measured … decimal(10, 2); Measured Forage Production in Pounds per Acre
percent_of_site_area … decimal(4,3); Percent of the Forage Inventory Site

Result Payload

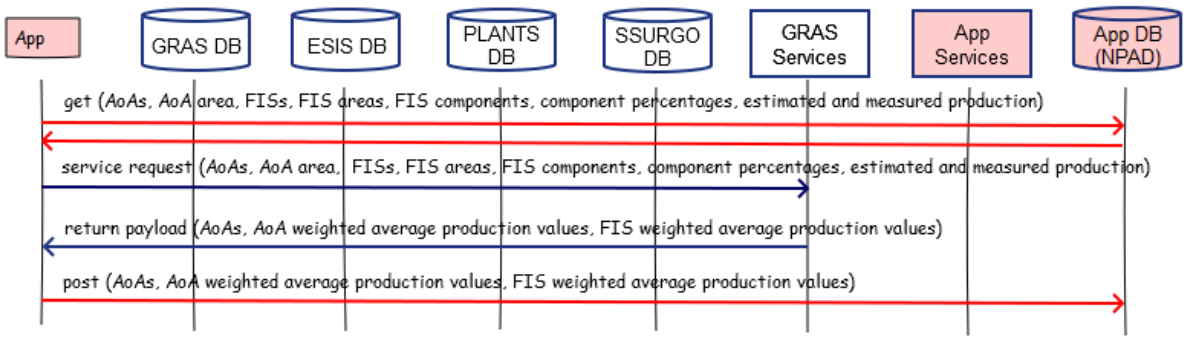
AoAId … integer, one; Area of Analysis Identifier
aoa_wt_avg_forage_prod_est … decimal(10, 2); Weighted Average Estimated Forage Production for the Area of Analysis in Pounds per Acre
aoa_wt_avg_forage_prod_measured … decimal(10, 2); Weighted Average Measured Forage Production for the Area of Analysis in Pounds per Acre

fis_id … integer, one or more per AoA; Forage Inventory Site Identifier
fis_wt_avg_forage_prod_est … decimal(10, 2); Weighted Average Estimated Forage Production for the Forage Inventory Site in Pounds per Acre
fis_wt_avg_forage_prod_measured … decimal(10, 2); Weighted Average Measured Forage Production for the Forage Inventory Site in Pounds per Acre

Transaction Data Sources

National Planning and Agreements Database (NPAD)
grazing_unit table
forage_inventory_site table
fis_component table

GRAS-1d: Calculate Weighted Average Estimated and Measured Forage Production Values



Component

1. **Calculate Weighted Average Forage Production Values (CalcWtAvProd)**
   1.1. Inputs
   AoA identifier
   AoA acres
   forage_inventory_site_id

forage_inventory_site_acres

component_id
forage_production_estimated
forage_production_measured
percent_of_site_area

1.2. Methods

For each AoAId (Area of Analysis)
For each fis_id (Forage Inventory Site in the AoA)
For each fis_component_id (component in the Forage Inventory Site)
#Calculate weighted average estimated production for FIS
component_weight = forage_production_estimated * percent_of_site_area
cumulative_wt_avg = cumulative_wt_avg + component_weight
If last component_id for the FIS
fis_wt_avg_forage_prod_est = cumulative_wt_avg

#Calculate weighted average measured production for FIS
component_weight = forage_production_measured * percent_of_site_area
cumulative_wt_avg = cumulative_wt_avg + component_weight
If last component_id for the FIS
fis_wt_avg_forage_prod_measured = cumulative_wt_avg

#Calculate weighted average estimated production for the AoA
fis_weight = fis_wt_avg_forage_prod_est * (forage_inventory_site_acres / aoa_acres)
cumulative_wt_avg = cumulative_wt_avg + fis_weight
If last forage_inventory_site_id for the land unit
aoa_wt_avg_forage_prod_est = cumulative_wt_avg

#Calculate weighted average measured production for the AoA
fis_weight = fis_wt_avg_forage_prod_measured * (forage_inventory_site_acres / aoa_acres)
cumulative_wt_avg = cumulative_wt_avg + fis_weight
If last forage_inventory_site_id for the land unit
aoa_wt_avg_forage_prod_measured = cumulative_wt_avg

1.3. Output

AoAID … one
aoa_wt_avg_forage_prod_est
aoa_wt_avg_forage_prod_measured

fis_id … one or more per AoA
fis_wt_avg_forage_prod_est
fis_wt_avg_forage_prod_measured

Service GRAS-2: Get Plant Growth Curves for Forage Inventory Site Components in an Area of Analysis (GetPGCurves)

Purpose: Find and return plant growth curves relevant to the forage inventory site (FIS) components of the grazing units (areas of analysis, or AoAs) in a grazing system.

A FIS contains soil components that occur within its boundary. A soil component from the Soil Survey Geographic (SSURGO) database, particularly for soils on grazing land, usually has an associated ecological site identifier, which can be mapped to an ecological site identifier in the Ecological Site Information System (ESIS). The identifier beginning with "R" refers to an ecological site on range or other land with non-forest natural plant communities. An identifier beginning with "F" refers to a site considered forest. And an identifier beginning with "G" refers to a forage suitability group on cultivated land (crop or pasture)

Both R and F ecological sites have plant communities, and each may have one or more growth curves. A forage suitability group (G site) may have one or more growth curves reflecting the forage species that can be cultivated on the site.

Growth curves allocate forage production by percentage to each month of the year. Growth curves are maintained in ESIS, and not stored in SSURGO.

An application using GRAS services usually will need plant growth curves after the user has selected an ecological site (ESD) or forage sutiability group (FSG) to represent a forage inventory site (FIS) component. Sometimes an ESD or FSG will not have growth curves, and as a backup the service also returns a list of default state and national growth curves.

Service Signature

Request Payload

AoAID … integer, one per request payload, Area of Analysis (AoA) Identifier
ofc_st_cty … character(5); State county code of office submitting request payload
fis_id … integer, one or more per AoA, Forage Inventory Site (FIS) identifier
fis_comp_id … integer, one or more per FIS, Forage Inventory Site Component Identifier
fis_es_id …. character varying(60), Ecological Site or Forage Suitability Group Identifier Associated to FIS; selected by application user to represent the FIS component
fis_plt_comm_id … numeric(2,0); one per es_id; Plant Community Identifier Associated to FIS component; user previously has selected a plant community to associate to the FIS component; if the site (es_id) is a FSG, the plant community identifier will be '99'

Result Payload

AoAID … one
fis_id … integer, one or more per AoA, Forage Inventory Site (FIS) Identifier
fis_comp_id … integer, one or more per FIS, Forage Inventory Site Component Identifier

es_id ... character varying(60), Ecological Site or Forage Suitability Group Identifier, one perFIS component
plant_community_id … numeric(2,0), one per FIS component, Plant Community Identifier
growth_curve_id … character varying(10), one or more for the plant community, Plant Growth Curve Identifier
growth_curve_name … character varying(100), Plant Growth Curve Name
growth_curve_description … character varying(1000), Plant Growth Curve Description
percent_production_jan … numeric(3,0), January Percent Forage Production
percent_production_feb … numeric(3,0), February Percent Forage Production
percent_production_mar … numeric(3,0), March Percent Forage Production
percent_production_apr … numeric(3,0), April Percent Forage Production
percent_production_may … numeric(3,0), May Percent Forage Production
percent_production_jun … numeric(3,0), June Percent Forage Production
percent_production_jul … numeric(3,0), July Percent Forage Production
percent_production_aug … numeric(3,0), August Percent Forage Production
percent_production_sep … numeric(3,0), September Percent Forage Production
percent_production_oct … numeric(3,0), October Percent Forage Production
percent_production_nov … numeric(3,0), November Percent Forage Production
percent_production_dec … numeric(3,0), December Percent Forage Production

state_county_code … numeric(2,0), one for the AoA, State County Code Identifier
growth_curve_id … character varying(10), one or more the state/county; Plant Growth Curve Identifier
growth_curve_name … character varying(100), Plant Growth Curve Name
growth_curve_description … character varying(1000), Plant Growth Curve Description
percent_production_jan … numeric(3,0), January Percent Forage Production
percent_production_feb … numeric(3,0), February Percent Forage Production
percent_production_mar … numeric(3,0), March Percent Forage Production
percent_production_apr … numeric(3,0), April Percent Forage Production
percent_production_may … numeric(3,0), May Percent Forage Production
percent_production_jun … numeric(3,0), June Percent Forage Production

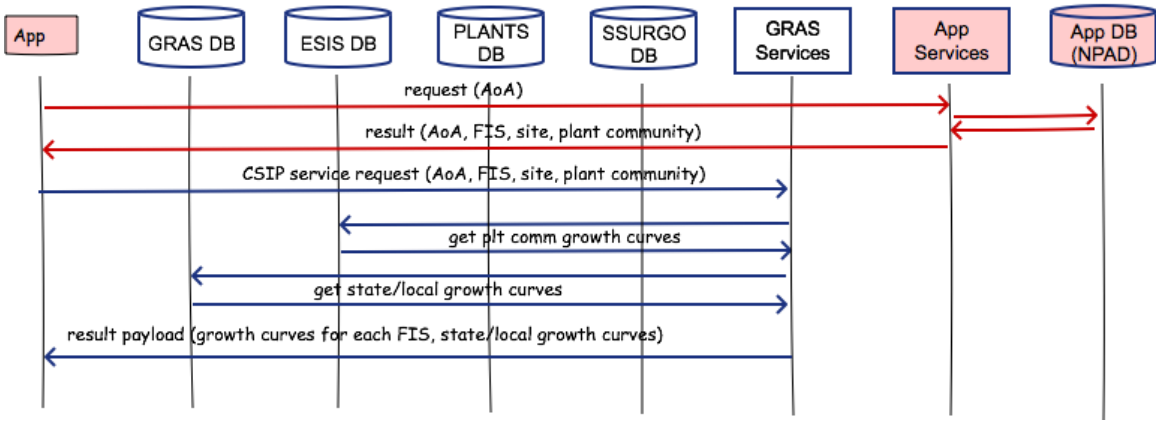
percent_production_jul … numeric(3,0), July Percent Forage Production
percent_production_aug … numeric(3,0), August Percent Forage Production
percent_production_sep … numeric(3,0), September Percent Forage Production
percent_production_oct … numeric(3,0), October Percent Forage Production
percent_production_nov … numeric(3,0), November Percent Forage Production
percent_production_dec … numeric(3,0), December Percent Forage Production

Reference Data Sources

Ecological Site Information System (ESIS) Data Mart
plant_growth_curves_lkp table
plant_communities_growth_curves table

GRAS Data Mart
d_plant_growth_curve_state_local table

GRAS-2: Get Growth Curves for Forage Inventory Sites



Components

1. **Get Plant Growth Curves (getPGCpcts)**
   1.1. Inputs

AoAId	1			
ofc-st_cty	48377			
fis_id	1	2		3
fis_comp_id	1	1	2	1
fis_es_id	R042XE694TX	R042XC249TX	R042XE281TX	R042XE694TX
fis_plt_comm_id	1	2	1	1

1.2. Data
- ESIS
  - plant_growth_curves_lkp
    - growth_curve_id
    - growth_curve_name
    - growth_curve_description
    - percent_production_jan
    - percent_production_feb
    - percent_production_mar
    - percent_production_apr
    - percent_production_may
    - percent_production_jun
    - percent_production_jul
    - percent_production_aug
    - percent_production_sep
    - percent_production_oct
    - percent_production_nov
    - percent_production_dec
  - plant_communities_growth_curve tables
    - es_type
    - es_mlra
    - es_mlru
    - es_site_number
    - es_state
    - plant_community_id
    - growth_curve_id
- GRAS Data Mart
  - d_plant_growth_curve_state_local table
    - growth_curve_id
    - state_county_code
    - growth_curve_name
    - growth_curve_description
    - percent_production_jan
    - percent_production_feb
    - percent_production_mar
    - percent_production_apr
    - percent_production_may
    - percent_production_jun
    - percent_production_jul
    - percent_production_aug
    - percent_production_sep
    - percent_production_oct
    - percent_production_nov
    - percent_production_dec

1.3. Methods

For the AoA

#Get plant growth curves for each FIS component in the AoA

For each fis_id in the AoA

For each fis_comp_id in the FIS

Select

growth_curve_id
growth_curve_name
growth_curve_description
plant_community_id
percent_production_jan
percent_production_feb
percent_production_mar
percent_production_apr
percent_production_may
percent_production_jun
percent_production_jul
percent_production_aug
percent_production_sep
percent_production_oct
percent_production_nov
percent_production_dec

Into fis_pgc_table
From ESIS plant_growth_curves_lkp
Inner Join plant_communities_growth_curves
where concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=fis_es_id and plant_community_id=fis_plt_comm_id

#Note: FSGs do not have plant communities, therefore plant_community_id is "99" for all FSGs having plant growth curves.

#Send to Output

Output data in fis_pgc table associated to fis_comp_id

#Set AoA state

aoa_state_cty = ofc_st_cty

#Get state/local plant growth curves

Select

growth_curve_id
growth_curve_name
growth_curve_description
state_county_code
percent_production_jan
percent_production_feb
percent_production_mar
percent_production_apr

percent_production_may
percent_production_jun
percent_production_jul
percent_production_aug
percent_production_sep
percent_production_oct
percent_production_nov
percent_production_dec

Into pgc_state_local table

From GRAS d_plant_growth_curve_state_local table **where** first two letters of d_plant_growth_curve_state_local.state_county_code == first two letters of aoa_state_cty

#Send to Output

Output data in pgc_state_local table

1.4. Outputs

AoAID … one per request
fis_id … one or more in AoA
fis_comp_id … one or more per FIS
es_id … one selected for FIS component
plant_community_id … one selected for FIS component
growth_curve_id … one or more per FIS component
growth_curve_name
growth_curve_description
percent_production_jan
percent_production_feb
percent_production_mar
percent_production_apr
percent_production_may
percent_production_jun
percent_production_jul
percent_production_aug
percent_production_sep
percent_production_oct
percent_production_nov
percent_production_dec
state_county_code … one per AoA
growth_curve_id … one or more per AoA
growth_curve_name
growth_curve_description
percent_production_jan
percent_production_feb
percent_production_mar
percent_production_apr
percent_production_may
percent_production_jun
percent_production_jul

percent_production_aug
percent_production_sep
percent_production_oct
percent_production_nov
percent_production_dec

Service GRAS-3: Get Forage Partition Profile (FPP) Templates for an Area of Analysis (GetFPPTemplates)

Purpose: Get forage partition profiles for one or more areas of analysis (AoAs) based on land use, land use modifier, and location state. The profiles contain periods where an AoA may be grazed by livestock, harvested for forage (e.g. hay, silage, etc.), or excluded from grazing. When grazed or harvested, a harvest efficiency is applied to the period so that remaining producton can be calculated.

Service Signature

Request Payload

- servicing_office_id … state county code of office submitting request payload
- AoA identifier … one or more
  - aoa_land_use… NRCS land_use_id integer (key to NRCS land use domain table: crop, range, pasture, other rural land, designated protected area, associated agricultural land)

Result Payload

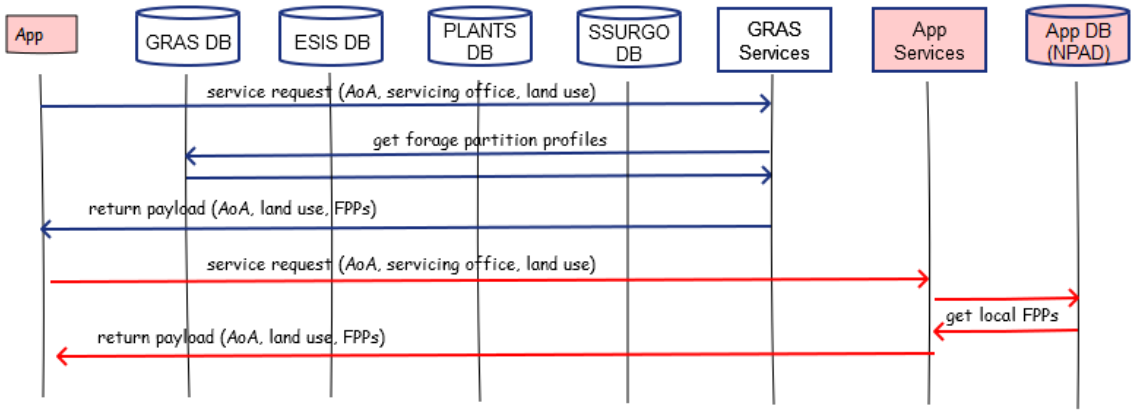
- AoA identifier … one or more
  - aoa_land_use
  - forage_partition_profile_id … one or more
    - state_county_code
    - plu_default_activity_ind
    - forage_partitition_profile_name
    - forage_partition_profile_description
    - start_date
    - end_date
    - forage_partition_profile_activity_id … one or more
      - forage_partition_activity_type_name
      - forage_partition_activity_type_display
      - harvest_efficiency_pct
      - calendar_start_day
      - calendar_end_day

Reference Data Sources

GRAS forage partition profile tables

- d_forage_partition_profile
- d_forage_partitiion_profile_activity
- d_forage_partition_profile_activity_type

GRAS-3: Get Forage Partition Profiles (FPPs)



Component

1. **Get Forage Partition Profile Parameters (getFPPparams)**
   1.1. Inputs
   servicing office_id
   AoA identifier … one or more
   aoa_land_use

   1.2. Data
   d_forage_partition_profile table
   forage_partition_profile_id
   state_county_code
   plu_default_activity_ind
   forage_partitition_profile_name
   forage_partition_profile_description
   start_date
   end_date

   1.3. Methods
   aoa_state = servicing_office_id
   this_day = today
   For each AoA
   #Get forage partition profiles (FPPs) for the AoA relevant to land use/modifier
   Select
   forage_partition_profile_id
   state_county_code
   plu_default_activity_ind
   forage_partitition_profile_name
   forage_partition_profile_description
   From GRAS d_forage_partition_profile table **where** first two letters of state_county_code == aoa_state and land_use_id == aoa_land_use this_day > start_date and <= end_date

Resulting aoa_fpp table
AoA Identifier
forage_partition_profile_id … one or more in the AoA
state_county_code … one per FPP
plu_default_activity_ind … one per FPP (yes or no)
forage_partitition_profile_name … one per FPP
forage_partition_profile_description … one per FPP

#Note: GRAS d_forage_partitiion_profile and d_forage_partition_profile activity tables may not be modeled correctly. Seems to make more sense to have land use and modifier in the d_forage_partition_profile table.

#Send to Output
Output data in aoa_fpp table for this AoA as JSON
#Get the set of activities in each FPP for the AoA
For each selected forage_partition_profile_id
current_fpp_id = this forage_partition_profile_id
Select
forage_partition_profile_activity_id
forage_partition_profile_id
forage_partition_activity_type_id
harvest_efficiency_pct
calendar_start_day
calendar_end_day
Into fpp_activity table
From GRAS d_forage_partition_profile_activity table **where**
forage_partition_profile_id == current_fpp_id

Resulting fpp_activity table
AoA identifier
forage_partition_profile_activity_id … one or more
forage_partition_profile_id
forage_partition_activity_type_id
harvest_efficiency_pct
calendar_start_day
calendar_end_day

#Get names for each activity in the FPP
Select
AoA identifier
fpp_activity.forage_partition_profile_activity_id
fpp_activity.forage_partition_profile_id
fpp_activity.forage_partition_activity_type_id
fpp_activity.harvest_efficiency_pct
fpp_activity.calendar_start_day
fpp_activity.calendar_end_day
d_forage_partition_activity_type.forage_partition_activity_type_name

d_forage_partition_activity_type.forage_partition_activity_type_display
Into fpp_activity_out table
From fpp_activity
Inner Join d_forage_partition_activity_type
On fpp_activity.forage_partition_profile_activity_id =
d_forage_partition_activity_type. forage_partition_profile_activity_id
Order By fis_mapunit.fis_id.cokey

Resulting fpp_activity_out table
AoA identifier
forage_partition_profile_activity_id … one or more
forage_partition_profile_id
forage_partition_activity_type_id
forage_partition_activity_type_name
forage_partition_activity_type_display
harvest_efficiency_pct
calendar_start_day
calendar_end_day

#Send to Output
Output data in fpp_activity_out table for this AoA as JSON

1.4. Outputs
AoA identifier … one or more
aoa_land_use
forage_partition_profile_id … one or more
state_county_code
plu_default_activity_ind
forage_partitition_profile_name
forage_partition_profile_description
start_date
end_date
forage_partition_profile_activity_id … one or more
forage_partition_activity_type_name
forage_partition_activity_type_display
harvest_efficiency_pct
calendar_start_day
calendar_end_day

Service GRAS-4a: Calculate Forage Area Adjustment Factors (CalcFAAFactors)

Purpose: Forage adjustment provides the ability to increase or decrease forage production and availability due to landscape conditions that limit the capture of forage, such as distance to water, slope, physical barriers, terrain or other site conditions.

From one or more water source point features in an area of analysis (AoA) create forage area adjustment FAA polygons and associated adjustment factors based on distance to water.

Intersect all AoA and FAA polygon geometry for all adjustment categories and then resolve FAA overlap using union geoprocessing. Calculate FAA areas and adjustment factors and return to requesting application.

Service Signature

Request Payload

AoAId … integer; one per request; Area of Analysis Identifier
- aoa_geometry … polygon, one set of coordinates; Area of Analysis Geometry
- water_faa_identifier … integer; one or more in AoA; Water Forage Adjustment Area Identifier
  - water_faa_geometry … point; Water Forage Adjustment Area Geometry
  - water_feature_factor_edits … boolean; Water Feature Factor Edit Flag
  - adj_factor_edit_id … integer, at least one and nor more than six, and consecutively numbered; Water Feature Factor Edit Identifier
    - min_extent_edit … integer; Minimum Extent in Feet
    - max_extent_edit … integer; Maximum Extent in Feet
    - adj_factor_edit … numeric(3,2); Forage Adjustment Factor; value almost always will be less than 1.00
- faa_id … integer, one or more in AoA; Forage Adjustment Area Identifier
  - faa_geometry … polygon, Forage Adjustment Area Geometry; can span AoA geometry
  - adjustment_category … character varying(50); Forage Adjustment Area Category; values are vegetation, slope, water, other
  - adjustment_factor … numeric(3,2); Forage Adjustment Factor; value almost always will be less than 1.00

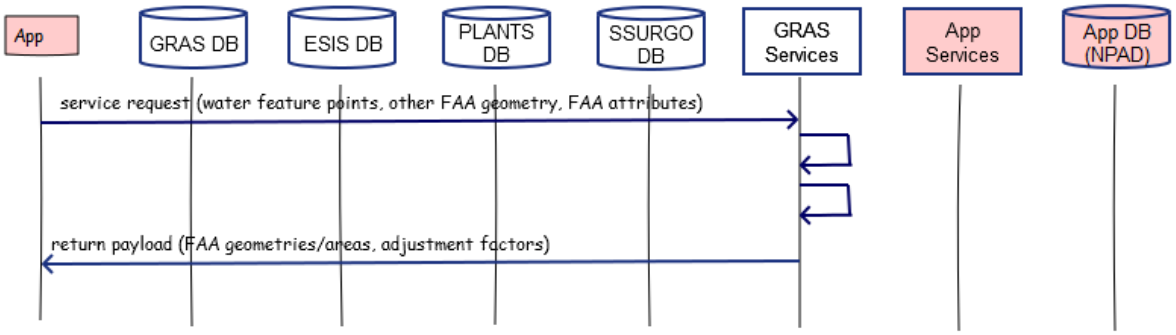
Result Payload

AoAId … integer; Area of Analysis Identifier
- aoa_faa_identifier … integer; one or more in AoA; Area of Analysis Forage Adjustment Area Identifier
  - aoa_faa_geometry … polygon; Area of Analysis Forage Adjustment Area Geometry
  - adjustment_area … numeric(6,1); Area of Analysis Forage Adjustment Area in Acres
  - adjustment_factor … numeric(3,2); Forage Adjustment Factor; value almost always will be less than 1.00

Reference Data Sources

GRAS Database
d_faa_water_adjustment_factor table

GRAS-4: Create Water Feature Forage Adjustment Areas (FAAs) and Calculate Adjustments



Component

1. **Compute Forage Area Adjustment Factors for One or More Water Features in an AoA (ComputeWater FAA)**

1.1. Inputs
AoA identifier
Water FAA identifier
Water FAA point geometry
adjustment_category … water

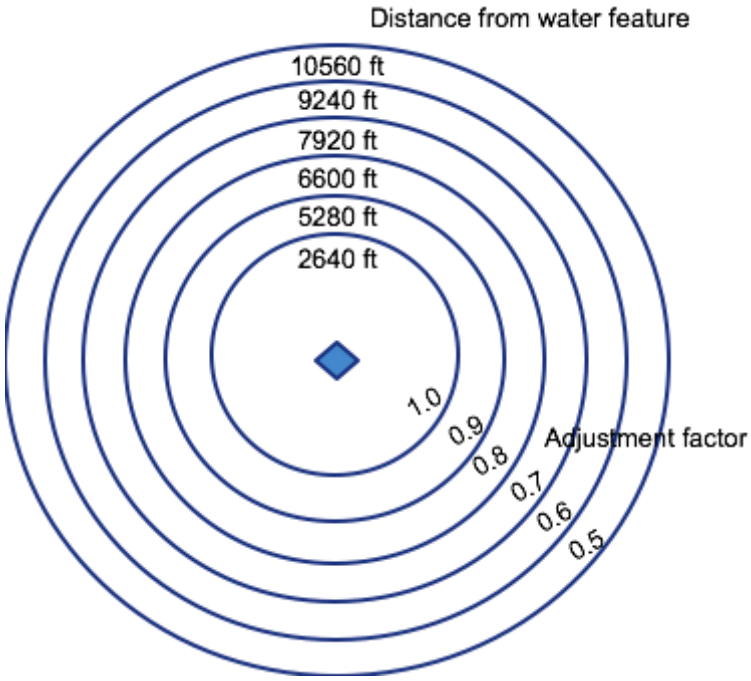
1.2. Data

d_faa_water_adjustment_factor

adj_factor_id	min_adj_extent	max_adj_extent	adj_factor
1	0	2640	1.0
2	2640	5280	0.9
3	5280	6600	0.8
4	6600	7920	0.7
5	7920	9240	0.6
6	9240	10560	0.5

1.3. GIS Operations / Method
For each AOA
#Create water FAA polygons
For each Water FAA identifier in the AoA
If water_feature_edits == yes
For each adj_factor_edit_id

Create circular water feature polygon around point water feature with minimum radius == min_extent_edit and maximum radius == max_extent_edit
adjustment_factor = adj_factor_edit
Else if water_feature_edits == no
For each adj_factor_id in d_faa_water_adjustment_factor table
Create circular water feature polygon around point water feature with minimum radius == min_adj_extent and maximum radius == max_adj_extent
adjustment_factor = adj_factor



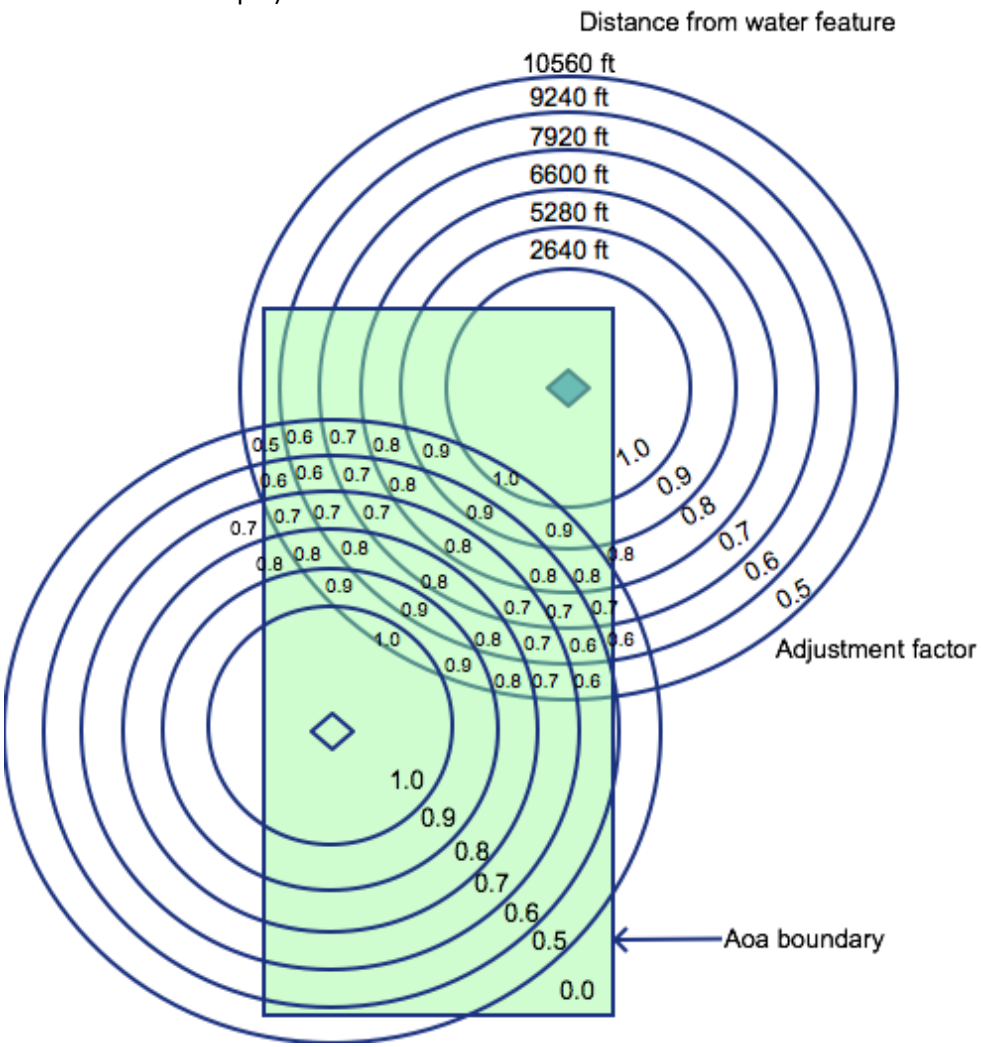
#Rather than creating water FAA polygons with each service call, GRAS d_faa_water_adjustment_factor table could already contain the geometry for each adjustment factor

Union AoA with FAA water feature polygons
Assign each new polygon an identifier (faa_water_id)
Calculate area of each polygon (adjustment_area)

#Compute adjustment factor for each new water FAA polygon
For each faa_water_id
#If polygon represents area beyond the greatest max_adj_extent (currently 10560) from any water feature in the AoA
If no contributing FAA water polygon
adjustment_factor = 0.0
Else

adjustment_factor = greatest adjustment factor of the contributing FAA water polygons

This will create a set of FAA water adjustment area polygons with adjustment factors computed and assigned as shown in following diagram (2 water feature example)



Alternate Logic:

For each water feature point (water_faa_identifier, water_faa_geometry)
    If water_feature_edits TRUE
        For each water_feature_edit_id (descending from highest id to lowest)
            this_buffer = buffer around point with buffer distance == max_extent_edit
            this_buffer_adj_factor = adj_factor_edit for this water_feature_edit_id

Alternate Logic (cont'd):

```
            If prev_buffer NULL
                Go to next water_feature_edit_id
                prev_buffer = this buffer
            Else if prev_buffer NOT NULL and not last iteration
                water_faa_polygon = prev_buffer DIFFERENCE this_buffer (donut polygon
                is created)
                water_faa_adj_factor = this_buffer_adj_factor
                water_faa_polygon = water_faa_polygon CLIP aoa_geometry
                Add water_faa_polygon, water_faa_adj_factor to feature set (layer)
                prev_buffer = this_buffer
            Else
                water_faa_polygon = this_buffer
                water_faa_adj_factor = this_buffer_adj_factor
                Add water_faa_polygon, water_faa_adj_factor to feature set (layer)
    Else if water_feature_edits FALSE
        For each water_faa_identifier
            For each adj_factor_id in d_faa_water_adjustment_factor table (descending
            from highest to lowest)
                this_buffer = buffer around point with buffer distance == max_adj_extent
                this_buffer_adj_factor = adj_factor for this adj_factor_id
                If prev_buffer NULL
                    Go to next adj_factor_id
                    prev_buffer = this buffer
                Else if prev_buffer NOT NULL and not last iteration
                    this_adjfact_polygon = prev_buffer DIFFERENCE this_buffer (donut
                    polygon is created)
                    water_faa_polygon = this_adjfact_polygon CLIP aoa_geometry
                    water_faa_adj_factor = this_buffer_adj_factor
                    Add water_faa_polygon, water_faa_adj_factor to feature set (layer)
                    prev_buffer = this_buffer
                Else
                    water_faa_polygon = this_buffer
                    water_faa_adj_factor = this_buffer_adj_factor
                    Add water_faa_polygon, water_faa_adj_factor to feature set (layer)
```

UNION all water_faa_polygon feature sets, which will create additional polygons where water feature polygons overlap; result is one water_faa_polygon feasture set (layer)
DISSOLVE resulting adjacent water_faa_polygons having same water_faa_adj_factor values
CLIP resulting water_faa_polygon feature set with aoa_geometry, trimming the water_faa_polygon feature set to occur within the AoA.

The next step is to incorporate other FAA areas of category Vegetation, Slope, Other.

1.4. Outputs
AoA identifier
faa_water_id
faa_water_geometry
adjustment_area
adjustment_factor

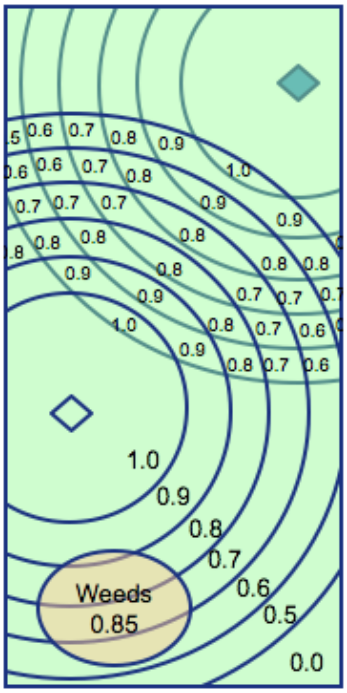
2. <u>Calculate Forage Area Adjustment Factors in an AoA (CalcAllFAA)</u>

2.1. Inputs
AoA identifier
#From ComputeWater FAA component (previous component this service)
faa_water_id … zero to many
faa_water_geometry
adjustment_area
adjustment_factor
#From request payload
FAA identifier … zero to many
FAA polygon geometry
adjustment_category … only vegetation, slope, other are relevant
adjustment_factor

2.2. GIS Operations / Methods
For each AoA
#Create and attribute full set of FAA polygons for all adjustment categories
#Two or more geometries can overlap
Union all faa_water_geometry and FAA polygon geometry in the AoA
For each resulting polygon
Assign each polygon an identifier
aoa_faa_id
Calculate adjustment_area

#Attribute new FAA polygons
For each new union polygon added to the set (e.g. the 4 new polygons created in the following example)
adjustment_factor = product(adjustment factors of contributing polygons to union)
#For example, adjustment factors for the 4 polygons created by union of water FAA with vegetation (weeds) FAA are:

0.9 * 0.85 = 0.765
0.8 * 0.85 = 0.680
0.7 * 0.85 = 0.595
0.6 * 0.85 = 0.510



Alternate Logic (cont'd):

For each faa_id
other_faa_polygon = aoa_faa_geometry
other_faa_adj_adfactor = adjustment_factor
Add other_faa_polygon, other_faa_adj_factor to feature set (layer)

UNION all other_faa_polygon feature sets, which will create additional polygons where other FAA feature polygons overlap; result is one other_faa_polygon feasture set (layer)
DISSOLVE resulting adjacent other_faa_polygons having same other_faa_adj_factor values
other_faa_adj_factor for resulting dissolved polygon = product(other_faa_adj_factor values of FAA polygons contributing to the dissolve)
CLIP resulting other_faa_polygon feature set with aoa_geometry, trimming the other_faa_polygon feature set to occur within the AoA.

UNION aoa_geometry, water_faa_polygon feature set, and other_faa_polygon feature set in that order into aoa_faa_polygon feature set (aoa_faa_identifier, aoa_faa_geometry, adjustment area, adjustment factor)
Where water FAA polygons and other FAA polygons overlap
For each new aoa_faa_polygon
adjustment factor = product(adjustment factors of contributing features to the new union polygon)
Where water and other FAA polygons do not overlap
For each such external aoa_faa_polygon
adjustment factor = 0
DISSOLVE adjacent FAA polygons having same adjustment_factor value

2.3. Output
AoA identifier … one or more
aoa_faa_id … one or more
AoA FAA geometry
adjustment_area
adjustment_factor

Service GRAS-4b: Get Forage Adjustment Categories (GetForAdjCat)

Purpose: Get and return a payload of the forage adjustment categories to enable the requesting application to associate forage adjustment reasons with forage adjustment polygons.

Service Signature

Request Payload

No data is passed into the service for processing other than requesting the service to run

Result Payload

forage_adjustment_category_id … smallint (1, 2, 3, 4), Forage Adjustment Type Identifier

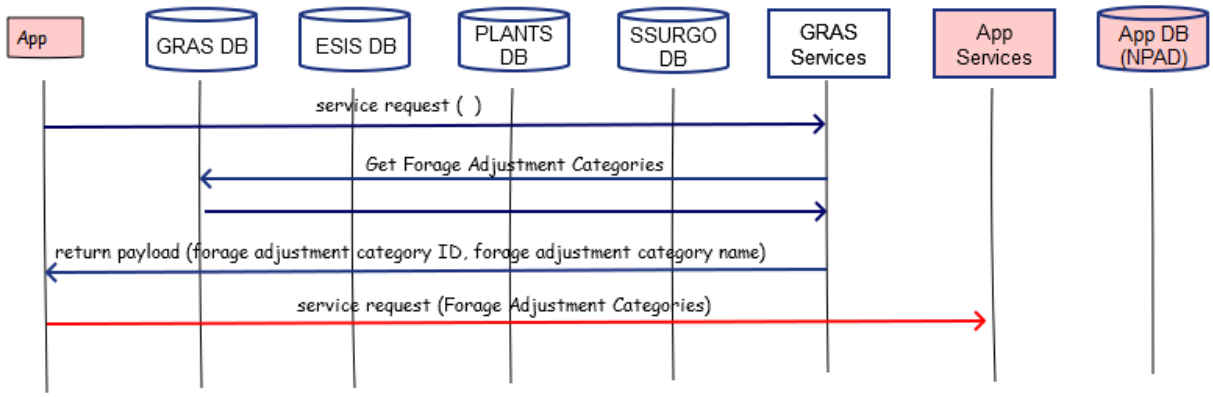
forage_adjustment_category_name … character varying (50); Forage Adjustment Name

Reference Data Sources

GRAS Database

d_forage_adjustment_category

GRAS-4b: Get Forage Adjustment Categories



Component

1. **Get Forage Adjustment Types (GetForAdjCat)**

1.1. Inputs

1.2. Method

Select

forage_adjustment_category_id

forage_adjustment_category_name

From GRAS d_forage_adjustment_category

#Send to Output
Output forage_adjustment_category_id, forage_adjustment_category_name

1.3. Outputs

forage_adjustment_category_id … smallint (1, 2, 3, 4), Forage Adjustment Type Identifier

forage_adjustment_category_name … character varying (50); Forage Adjustment Name

Service GRAS-5: Calculate Adjusted Annual Forage Production for an Area of Analysis (CalcProdAnnual)

Purpose: Overlay forage inventory site (FIS) and forage adjustment area (FAA) geometries to create FAA polygons for each FIS in the area of analysis (AoA). Calculate weighted average annual adjusted production for FIS components, FISs, and AoAs. Application users may enter measured production either by FIS or by FIS components (but not both). The calculation method in this service triggers on whether the FIS is described to the FIS component level.

Service Signature

Request Payload

AoAId … integer, one in the request; Area of Analysis Identifier
- aoa_geometry … one set of geospatial coordinates for the AoA (one polygon); Area of Analysis Geometry
- aoa_faa_id … integer, one or more in the AoA; Forage Area Adjustment Identifier; from FAAs created in GRAS-4
  - faa_geometry … one set of coordinates for the adjustment area (one polygon); Forage Area Adjustment Geometry
  - adjustment_factor … numeric(3,2); Forage Area Adjustment Factor; e.g. 0.90
- fis_id … integer, … one or more in the AoA; Forage Inventory Site Identifier
  - fis_geometry … one set of geospatial coordinates for the FIS; Forage Inventory Site Geometry
  - fis_component_id … integer, one or more in the FIS; Forage Inventory Site Component Identifier
    - fis_component_pct … numeric(3,0); Component Percent of the Forage Inventory Site
    - fis_component_prod … numeric(5,0); Measured Annual Production for the Forage Inventory Site Component in Pounds per Acre

Result Payload

AoAId … one; Area of Analysis Identifier
- aoa_adj_prod_annual … numeric(5,0); Adjusted Measured Annual Production for the Area of Analysis in Pounds per Acre; this value gets saved to NPAD
- fis_id … integer, one or more in the AoA; Forage Inventory Site Identifier
  - fis_adj_prod_annual … numeric(5,0); Adjusted Measured Annual Production for the Forage Inventory Site in Pounds per Acre; this value gets saved to NPAD
  - fis_component_id …. one or more in the FIS; Forage Inventory Site Component Identifier
    - fis_comp_adj_prod_annual … numeric(5,0); Adjusted Measured Annual Production for the Forage Inventory Site Component in Pounds per Acre; this value gets saved to NPAD

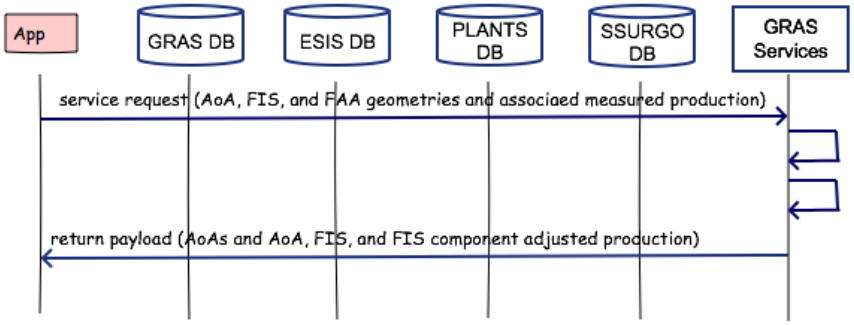
Reference Data Sources

None

Components

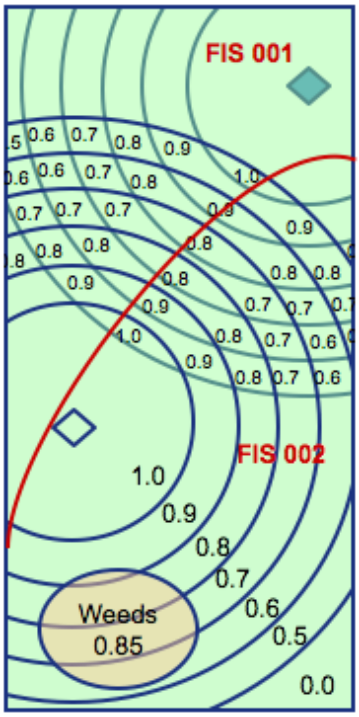
1. **Calculate Unadjusted Measured Forage Production for FIS Adjustment Areas (CalcFisFaaProd)**
   1.1. Inputs
   AoA identifier … one or more
   AoA polygon geometry
   FAA identifier … one or more
   FAA polygon geometry
   adjustment_factor
   FIS identifier … one or more
   FIS polygon geometry
   fis_production
   fis_component_id … one or more
   fis_component_pct
   fis_component_prod … measured production

GRAS-5: Calculate Adjusted Annual Forage Production for AoAs, FISs, and

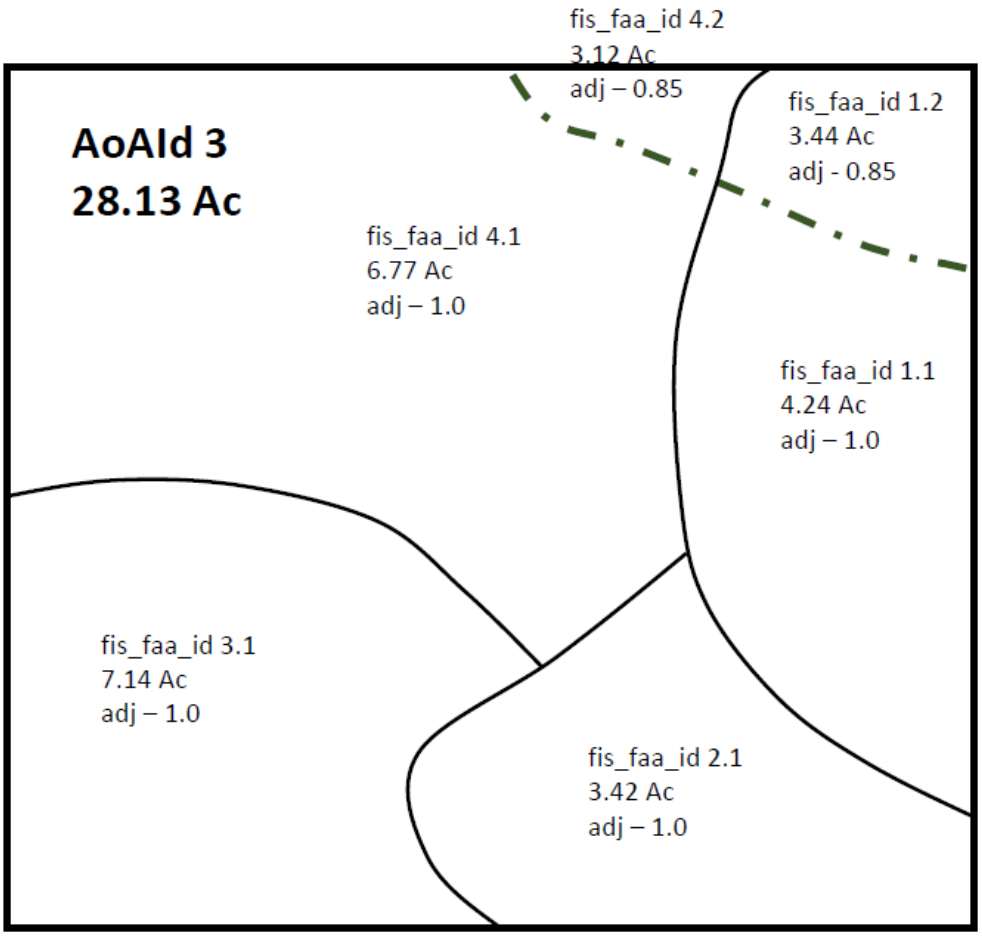


1.2. GIS Operations
For each AOA
#Compute AoA and FIS areas from their geometries
aoa_area = area(AoA geometry)
For each FIS in the AoA
fis_area = area(FIS geometry)

#Create forage inventory site (FIS) adjustment area (FAA) polygons; filling up the AoA with no gaps
Union all FIS geometry and FAA geometry to create FIS FAA polygons for the AoA



Another example:



Resulting data from intersection of

AoAId	aoa_area	fis_id	fis_area	fis_faa_id	fis_faa_area	adjustment_factor	fis_component_id	fis_component_pct	fis_component_prod
3	28.13	1	7.68	1.1	4.24	1.00	1	0.6	1000
							2	0.2	1000
							3	0.2	1500
				1.2	3.44	0.85	1	0.6	1000
							2	0.2	1000
							3	0.2	1500
		2	3.42	2.1	3.42	1.00	1	1	1000
		3	7.14	3.1	7.14	1.00	1	0.55	1000
							2	0.45	2000
		4	9.89	4.1	6.77	1.00	1	1	1500
				4.2	3.12	0.85			

1.3. Method

For the AoA

For each fis_id

For each fis_faa_id in the FIS

For each fis_component_id

fis_faa_comp_prod = fis_component_prod * adjustment_factor
sub_comp_adj_prod = fis_faa_comp_prod * (fis_faa_area / fis_area)
fis_comp_adj_prod_annual = fis_comp_adj_prod_annual +
sub_comp_adj_prod
sub_fis_adj_prod_annual = fis_comp_adj_prod_annual *
fis_component_pct
fis_adj_prod_annual = fis_adj_prod_annual + sub_fis_adj_prod_annual
sub_aoa_adj_prod = fis_adj_prod_annual * fis_area / aoa_area
aoa_adj_annual_prod = aoa_adj_annual_prod + sub_aoa_adj_prod

1.4. Output to service results payload
AoA identifier … one
aoa_adj_prod_annual
fis_id … one or more
fis_adj_prod_annual
fis_component_id… one, two, or three
fis_comp_adj_prod_annual

Service GRAS-6: Calculate Monthly and Daily Forage Production for an Area of Analysis (CalcProdMonthlyDaily)

Purpose: Calculate weighted average **monthly** and **daily** forage production for each AoA. From each monthly production, divide by the number of days in the month to get daily production.

Service Signature

Request Payload

- AoA identifier … one or more
  - forage_inventory_type … basic or detailed
  - FIS identifier … one or more in the AoA
    - fis_area
    - fis_component_id … one or more in the FIS
      - fis_component_pct
      - fis_comp_adj_prod_annual
      - comp_pct_prod_jan
      - comp_pct_prod_feb
      - comp_pct_prod_mar
      - comp_pct_prod_apr
      - comp_pct_prod_may
      - comp_pct_prod_jun
      - comp_pct_prod_jul
      - comp_pct_prod_aug
      - comp_pct_prod_sep
      - comp_pct_prod_oct
      - comp_pct_prod_nov
      - comp_pct_prod_dec

Result Payload

- AoA … one or more
  - aoa_prod_jan
  - aoa_prod_feb
  - aoa_prod_mar
  - aoa_prod_apr
  - aoa_prod_may
  - aoa_prod_jun
  - aoa_prod_jul
  - aoa_prod_aug
  - aoa_prod_sep
  - aoa_prod_oct
  - aoa_prod_nov
  - aoa_prod_dec

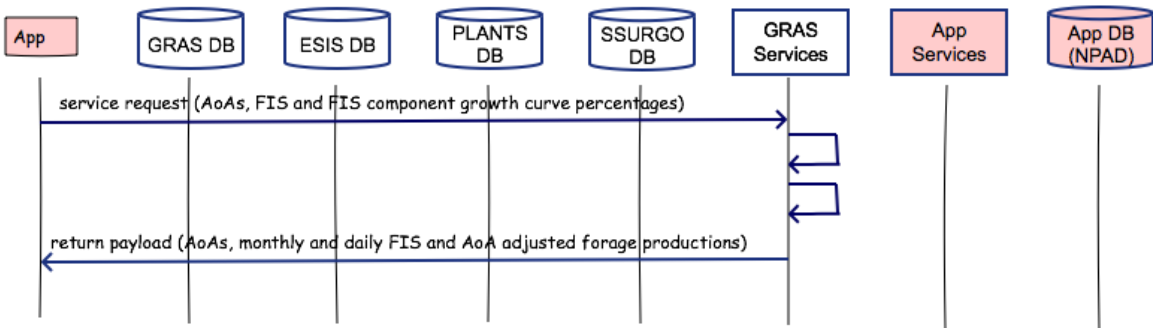
  - aoa_prod_jan_daily
  - aoa_prod_feb_daily
  - aoa_prod_feb_leap_daily
  - aoa_prod_mar_daily

aoa_prod_apr_daily
aoa_prod_may_daily
aoa_prod_jun_daily
aoa_prod_jul_daily
aoa_prod_aug_daily
aoa_prod_sep_daily
aoa_prod_oct_daily
aoa_prod_nov_daily
aoa_prod_dec_daily

Reference Data Sources

None

GRAS-6: Calculate Monthly and Daily Forage Production for AoAs



Components

1. **Calculate MonthlyForage Production Based on Plant Growth Period Percentage (CalcProdMonthly)**
   1.1. Input
   AoA identifier … one or more
   forage_inventory_type … basic or detailed
   FIS identifier … one or more in the AoA
   fis_area
   fis_component_id … one or more in the FIS
   fis_component_pct
   fis_comp_adj_prod_annual
   comp_pct_prod_jan
   comp_pct_prod_feb
   comp_pct_prod_mar
   comp_pct_prod_apr
   comp_pct_prod_may
   comp_pct_prod_jun
   comp_pct_prod_jul

comp_pct_prod_aug
comp_pct_prod_sep
comp_pct_prod_oct
comp_pct_prod_nov
comp_pct_prod_dec

1.2. Method

For each AoA

For each FIS in the AoA

#Calculate monthly AoA production

For each fis_component_id in the FIS

aoa_prod_jan = aoa_prod_jan + (fis_comp_adj_prod_annual * comp_pct_prod_jan * fis_area * fis_component_pct)

aoa_prod_feb = aoa_prod_feb + (fis_comp_adj_prod_annual * comp_pct_prod_feb * fis_area * fis_component_pct)

aoa_prod_mar = aoa_prod_mar + (fis_comp_adj_prod_annual * comp_pct_prod_mar * fis_area * fis_component_pct)

aoa_prod_apr = aoa_prod_apr + (fis_comp_adj_prod_annual * comp_pct_prod_apr * fis_area * fis_component_pct)

aoa_prod_may = aoa_prod_may + (fis_comp_adj_prod_annual * comp_pct_prod_may * fis_area * fis_component_pct)

aoa_prod_jun = aoa_prod_jun + (fis_comp_adj_prod_annual * comp_pct_prod_jun * fis_area * fis_component_pct)

aoa_prod_jul = aoa_prod_jul + (fis_comp_adj_prod_annual * comp_pct_prod_jul * fis_area * fis_component_pct)

aoa_prod_aug = aoa_prod_aug + (fis_comp_adj_prod_annual * comp_pct_prod_aug * fis_area * fis_component_pct)

aoa_prod_sep = aoa_prod_sep + (fis_comp_adj_prod_annual * comp_pct_prod_sep * fis_area * fis_component_pct)

aoa_prod_oct = aoa_prod_oct + (fis_comp_adj_prod_annual * comp_pct_prod_oct * fis_area * fis_component_pct)

aoa_prod_nov = aoa_prod_nov + (fis_comp_adj_prod_annual * comp_pct_prod_nov * fis_area * fis_component_pct)

aoa_prod_dec = aoa_prod_dec + (fis_comp_adj_prod_annual * comp_pct_prod_dec * fis_area * fis_component_pct)

1.3. Output to results payload
AoA identifier… one or more
aoa_prod_jan
aoa_prod_feb
aoa_prod_mar
aoa_prod_apr
aoa_prod_may
aoa_prod_jun
aoa_prod_jul
aoa_prod_aug
aoa_prod_sep
aoa_prod_oct
aoa_prod_nov
aoa_prod_dec

2. <u>Calculate Daily Forage Production Based on Plant Growth Period Percentage (CalcProdDaily)</u>

2.1. Input from previous component
AoA identifier… one or more
aoa_prod_jan
aoa_prod_feb
aoa_prod_mar
aoa_prod_apr
aoa_prod_may
aoa_prod_jun
aoa_prod_jul
aoa_prod_aug
aoa_prod_sep
aoa_prod_oct
aoa_prod_nov
aoa_prod_dec

2.2. Method
For each AoA
#Calculate daily AoA production (by month) and send to output
aoa_prod_jan_daily = aoa_prod_jan / 31
aoa_prod_feb_daily = aoa_prod_feb / 28
aoa_prod_feb_leap_daily = aoa_prod_feb / 29
aoa_prod_mar_daily = aoa_prod_mar / 31
aoa_prod_apr_daily = aoa_prod_apr / 30
aoa_prod_may_daily = aoa_prod_may / 31
aoa_prod_jun_daily = aoa_prod_jun / 30
aoa_prod_jul_daily = aoa_prod_jul / 31
aoa_prod_aug_daily = aoa_prod_aug / 31
aoa_prod_sep_daily = aoa_prod_sep / 30
aoa_prod_oct_daily = aoa_prod_oct / 31

aoa_prod_nov_daily = aoa_prod_nov / 30
aoa_prod_dec_daily = aoa_prod_dec / 31

2.3. Output to results payload

Aoa identifier … one or more

aoa_prod_jan_daily
aoa_prod_feb_daily
aoa_prod_feb_leap_daily
aoa_prod_mar_daily
aoa_prod_apr_daily
aoa_prod_may_daily
aoa_prod_jun_daily
aoa_prod_jul_daily
aoa_prod_aug_daily
aoa_prod_sep_daily
aoa_prod_oct_daily
aoa_prod_nov_daily
aoa_prod_dec_daily

Service GRAS-7: Get Animal Attributes for Herd Description (GetHerdDescrAttrib)

Purpose: Get and return a payload of animal attributes to enable the requesting application to describe one or more grazing herds, including animal kind, class, gender, intake requirement, gestation period, and other attributes necessary to calculate demand for forage through a grazing schedule.

Service Signature

Request Payload

No data is passed into the service for processing other than requesting the service to run

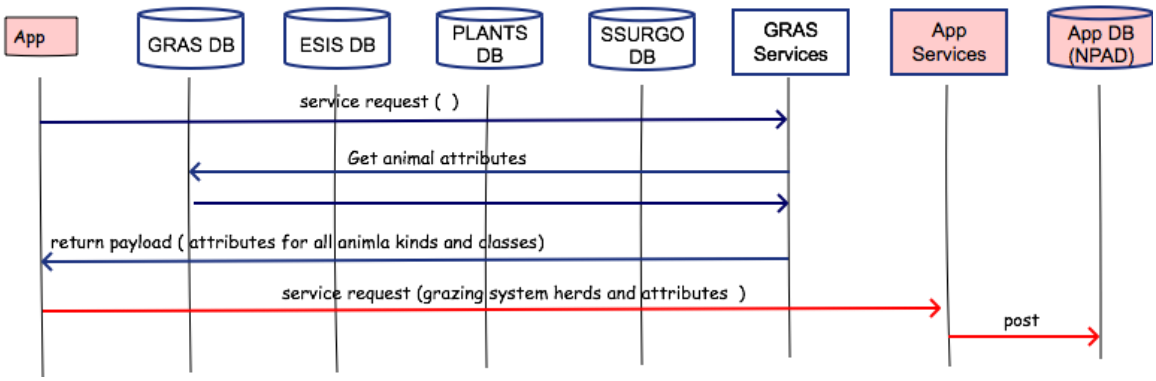
Result Payload

animal_unit_id …all animal units in the table
- animal_kind
- animal_class
- animal_gender
- animal_growth_category
- animal_default_aue
- animal_default_weight
- animal_avg_dailyintake_pct
- animal_default_gestatation_period_days

Reference Data Sources

GRAS Database
- d_animal_unit
- d_animal_kind

GRAS-7: Get Animal Attributes for Herd Description



Component

1. **Get Animal Unit Attributes for Animal Group and Herd Description (getAnimalAttrib)**

1.1. Inputs

1.2. Method

Join d_animal_unit and d_animal_kind tables on animal_kind

Select

animal_unit_id
animal_kind
animal_class
animal_gender
animal_growth_category
animal_default_aue
animal_avg_dailyintake_pct
animal_default_gestatation_period_days
(animal_default_aue * 1000) **As** animal_default_weight

Into animal_attributes

From GRAS d_animal_unit x d_animal_kind table join where this_day >= start_date and <= end_date

Resulting animal_attributes table data elements

animal_unit_id … many
animal_kind
animal_class
animal_gender
animal_growth_category
animal_default_aue
animal_avg_dailyintake_pct
animal_default_gestatation_period_days
animal_default_weight

#Send to Output

Output data in compc_output_table for this FIS cokey and fis_id as JSON

1.3. Outputs

animal_unit_id …all active animal units in the d_animal_unit table

animal_kind
animal_class
animal_gender
animal_growth_category
animal_default_aue
animal_default_weight
animal_avg_dailyintake_pct
animal_default_gestatation_period_days

Service GRAS-8: Compute Breeder and Offspring Attributes (CompReprodAttributes)

Purpose: From application inputs, compute default begin and end breeding exposure dates, default number of offspring, and a back calculation of breeding efficiency.

Most of the processes involved with defining a male breeder group, an offspring group, and their forage demand will be done as application services and not GRAS services. This service involves methods that get data from the GRAS data mart.

Service Signature

Request Payload

Herd identifier
- Maternal Group identifier
  - maternal_animal_kind … from NPAD
  - maternal_animal_class … from NPAD
  - number_females … from NPAD
  - date_into_herd … from NPAD
  - date_out_of_herd … from NPAD
  - target_offspring_date … user entered
  - gestation_period_days … user entered
  - birthing_efficiency … percent, user entered
  - twinning_efficiency … percent, user entered
  - offspring_number ... user entered
- male_animal_kind … any of d_animal_unit.animal_kind values
- breeding_exposure_days … user entered

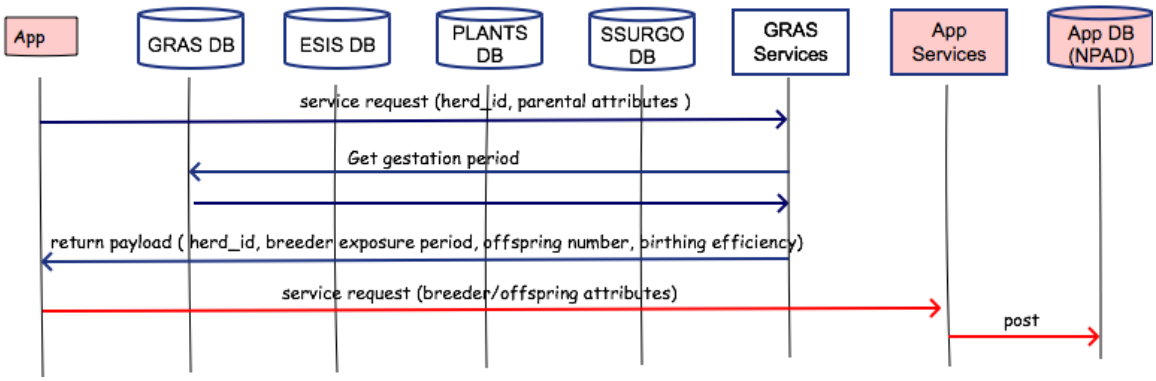
Result Payload

Herd identifier
- default_begin_exposure_date
- default_end_exposure_date
- default_offspring_number
- backcalc_birthing_efficiency

Reference Data Sources

GRAS database
- animal_unit table
- animal_kind table

GRAS-8: Compute Breeder and Offpring Attributes



Component

1. **Compute Breeder and Offspring Attributes for Reproductive Groups in the Herd (CompBreederOffspring)**
   1.1. Inputs

   1.2. Data
   d_animal_unit table
   animal_kind
   animal_default_gestation_period_days

   1.3. Methods
   If male_animal_kind == female_animal_kind
   #Validate that either birthing efficiency or offspring number is populated
   If birthing_efficiency AND offspring_number > 0 **OR** birthing_efficiency AND offspring_number = 0
   Return error message “Request should contain either offspring number or birthing efficiency”.

   #Compute default begin exposure date
   default_begin_exposure_date = target_offspring_date – gestation_period_days

   #Compute default end exposure date
   default_end_exposure_date = default_begin_exposure_date + breeding_exposure_days

   #Verify exposure dates are within Herd Start and End dates for the group
   If default_begin_exposure_date < date_into_herd (i.e. earlier) AND/OR default_end_exposure_date > date_out_of_herd (i.e. later)
   Return error message alerting user that exposure dates exceed the Start and/or End Herd dates for group

#Compute default number of offspring
If no offspring_number in request payload
default_offspring_number = round(number_females * (birthing_efficiency + (birthing_efficiency * twinning_efficiency)),0)
Else if offspring_number in request payload exists
#Back calculate birthing efficiency
backcalc_birthing_effciency = round((offspring_number / number_females) / (1 + twinning_efficiency),1)

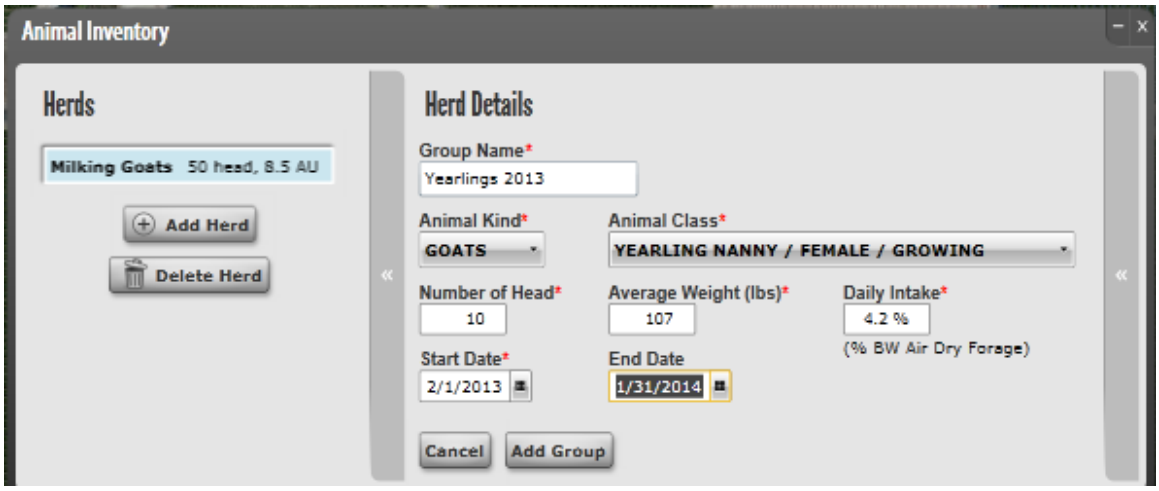
1.4. Output
Herd identifier
Maternal Group identifier
default_begin_exposure_date
default_end_exposure_date
default_offspring_number
backcalc_birthing_efficiency

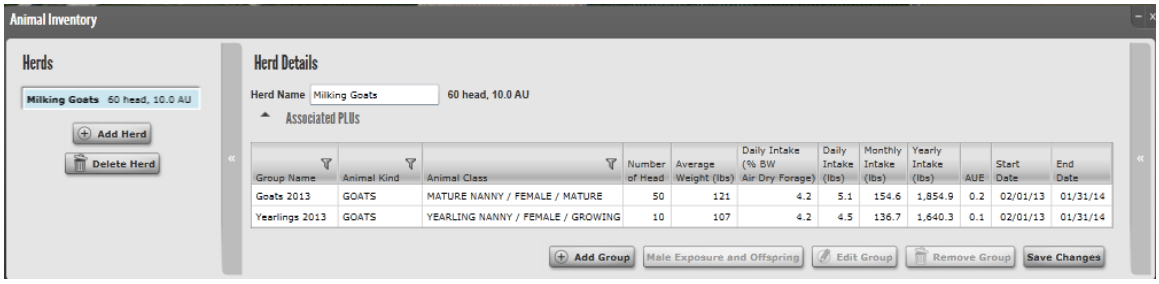
Service GRAS-9: Calculate Animal Herd Forage Demand (CalcAnimalHerdDemand)

Purpose: Calculate daily, monthly, and yearly animal demand for forage by animal group within a herd based on grazing system start and end dates. The benchmark condition will only be for a grazing period of one year. For developing alternatives, user identifies the number of years in planning horizon).

Calculate monthly demand by animal group and herd based on group start and end dates. This service updates demand as groups are added to a herd, or edited. The service also calculates group and herd totals for the monthly demand by group and herd reports.

The user enters/edits data about an animal group added to a herd. Upon update the service calculates daily, monthly, and yearly demand as well as animal unit equivalent (AUE) on a per head basis. The UI below is an example only.





Animal group start and end date factor into computing monthly demand for forage by herd and by groups within the herds of the grazing system.

Monthly Demand

By Herd

(lbs air-dry forage)

	Month												
Herd	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Herd 1	86257.5	77910	86257.5	86175	90487.5	91575	94627.5	92827.5	86175	83700	81000	83700	1040692.5
Herd 2	5580	5040	5580	5400	4140	0	0	1800	5400	5580	5400	5580	49500
Herd 3	0	0	0	0	0	0	0	0	25200	48825	47250	0	121275
Herd 4	0	0	15300	27000	27900	27000	27900	12600	0	0	0	0	137700
Monthly Total	91837.5	82950	107137.5	118575	122527.5	118575	122527.5	107227.5	116775	138105	133650	89280	1349167.5

Monthly Demand

By Group

(lbs air-dry forage)

		Month												
Herd	Group	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Total
Herd 1														
	Cows & Calves	69750	63000	69750	67500	69750	67500	69750	69750	67500	69750	67500	69750	821250
	Cull Cows - with calf	5347.5	4830	5347.5	5175	5347.5	5175	5347.5	5347.5	5175	0	0	0	47092.5
	Cull Cows - no calf	5580	5040	5580	0	0	0	0	0	0	0	0	0	16200
	Cows - No Calf - Keep	5580	5040	5580	5400	5580	5400	5580	5580	5400	5580	5400	5580	65700
	Breeding Heifer	0	0	0	8100	8370	8100	8370	8370	8100	8370	8100	8370	74250
	Bulls - Herd 1	0	0	0	0	1440	5400	5580	3780	0	0	0	0	16200
Herd 2														
	Bulls - Herd 1	5580	5040	5580	5400	4140	0	0	1800	5400	5580	5400	5580	49500
Herd 3														
	Weaned Calves	0	0	0	0	0	0	0	0	25200	48825	47250	0	121275
Herd 4														
	Purchased Stocker	0	0	15300	27000	27900	27000	27900	12600	0	0	0	0	137700
	Monthly Total	91837.5	82950	107137.5	118575	122527.5	118575	122527.5	107227.5	116775	138105	133650	89280	1349167.5

Service Signature

Request Payload

grazing_system_start_date … e.g. January 1, 2016 (in suitable date format)
grazing_system_end_date … e.g. December, 2020 (in suitable date format)

#Herd data for calculating daily, monthly, and yearly animal demand for the entire grazing system period.
herd_id … one or more
    animal_group_id … one or more
        animal_unit_id
        number_of_head
        avg_weight_lbs
        avg_daily_intake_pct
        date_into_herd
        date_out_of_herd

Result Payload

grazing_system_year … one or more in the grazing system
    herd_id … one or more
        animal_group_id … one or more
            #Animal group intake rates and animal unit equivalents
            animal_daily_intake_lbs
            animal_monthly_intake_lbs

animal_yearly_intake_lbs
group_daily_intake_lbs
animal_aue

#Animal group forage demand
jan_group_demand
feb_group_demand
mar_group_demand
apr_group_demand
may_group_demand
jun_group_demand
jul_group_demand
aug_group_demand
sep_group_demand
oct_group_demand
nov_group_demand
dec_group_demand
total_group_demand

#Herd forage demand
jan_herd_demand
feb_herd_demand
mar_herd_demand
apr_herd_demand
may_herd_demand
jun_herd_demand
jul_herd_demand
aug_herd_demand
sep_herd_demand
oct_herd_demand
nov_herd_demand
dec_herd_demand
total_herd_demand

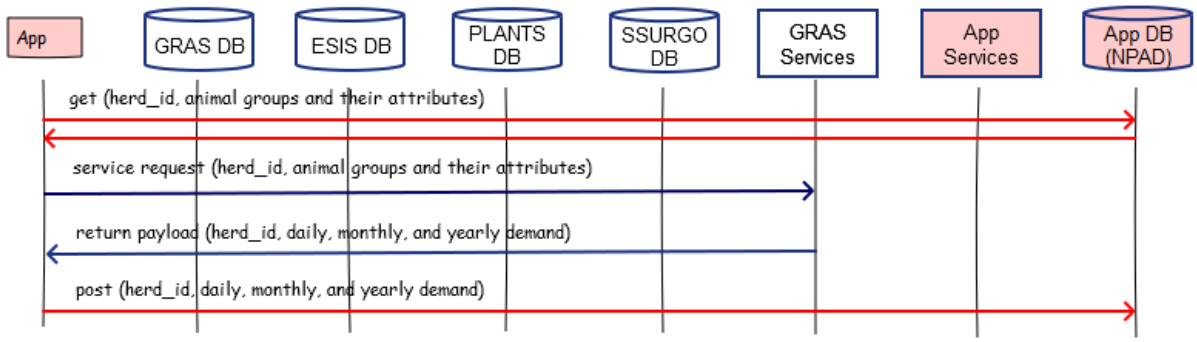
#Grazing system forage demand
jan_system_demand
feb_system_demand
mar_system_demand
apr_system_demand
may_system_demand
jun_system_demand
jul_system_demand
aug_system_demand
sep_system_demand
oct_system_demand
nov_system_demand
dec_system_demand

total_system_demand

Reference Data Sources

None used for this service

GRAS-9: Calculate Animal Herd Forage Demand



Component

1. **Calculate Daily, Monthly, and Yearly Herd Demand (CalcHerdDemand))**

1.1. Inputs

grazing_system_start_date … e.g. January 1, 2016 (in suitable date format)
grazing_system_end_date … e.g. December, 2020 (in suitable date format)
herd_id … one or more
    animal_group_id … one or more
        animal_unit_id
        number_of_head
        avg_weight_lbs
        avg_daily_intake_pct
        date_into_herd
        date_out_of_herd

1.2. Methods

#Set grazing system year
grazing_system_year = year in the grazing system (e.g. 2016)

For each grazing_system_year … one or more in the grazing system
    For each herd_id … one or more in the grazing system
        For each animal_group_id … one or more in the herd
            #Calculate daily, monthly, and yearly forage demand per head in group
            animal_daily_intake_lbs = avg_weight_lbs * avg_daily_intake_pct
            animal_monthly_intake_lbs = animal_daily_intake_lbs * (365 / 12)
            animal_yearly_intake_lbs = animal_daily_intake_lbs * 365

            #Calculate daily forage demand for the group
            group_daily_intake_lbs = animal_daily_intake_lbs * number_of_head

```
#Calculate animal unit equivalents (AUEs) per head and per group
animal_aue = (avg_weight_lbs * avg_daily_intake_pct * 365) / 10950

#Calculate monthly group forage demand
prev_month_demand = 0.00
For each day in a year
    this_day = this iteration's date
    If this_day >= date_into_herd and <= date_out_of_herd
        cumulative_demand = cumulative_demand +
        group_daily_intake_lbs
    If this_day == January 31st
        jan_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if leap year and this_day == February 29th
        feb_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == February 28th
        feb_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == March 31st
        mar_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == April 30th
        apr_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == May 31st
        may_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == June 30th
        jun_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == July 31st
        jul_group_demand = cumulative_demand – prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == August 31st
        aug_group_demand = cumulative_demand –
        prev_month_demand
        prev_month_demand = cumulative_demand
    Else if this_day == September 30th
```

sep_group_demand = cumulative_demand –
prev_month_demand
prev_month_demand = cumulative_demand
Else if this_day == October 31st
oct_group_demand = cumulative_demand –
prev_month_demand
prev_month_demand = cumulative_demand
Else if this_day == November 30th
nov_group_demand = cumulative_demand –
prev_month_demand
prev_month_demand = cumulative_demand
Else if this_day == December 31st
dec_group_demand = cumulative_demand –
prev_month_demand
total_group_demand = cumulative_demand

#Send to output for this group
Output jan_group_demand, feb_group_demand, mar_group_demand, apr_group_demand, may_group_demand, jun_group_demand, jul_group_demand, aug_group_demand, sep_group_demand, oct_group_demand, nov_group_demand, dec_group_demand, total_group_demand

#Update monthly herd demand
jan_herd_demand = jan_herd_demand + jan_group_demand
feb_herd_demand = feb_herd_demand + feb_group_demand
mar_herd_demand = mar_herd_demand + mar_group_demand
apr_herd_demand = apr_herd_demand + apr_group_demand
may_herd_demand = may_herd_demand + may_group_demand
jun_herd_demand = jun_herd_demand + jun_group_demand
jul_herd_demand = jul_herd_demand + jul_group_demand
aug_herd_demand = aug_herd_demand + aug_group_demand
sep_herd_demand = sep_herd_demand + sep_group_demand
oct_herd_demand = oct_herd_demand + oct_group_demand
nov_herd_demand = nov_herd_demand + nov_group_demand
dec_herd_demand = dec_herd_demand + dec_group_demand
total_herd_demand = total_herd_demand + total_group_demand

#Send to output for this herd
If last group in the herd
Output jan_herd_demand, feb_herd_demand, mar_herd_demand, apr_herd_demand, may_herd_demand, jun_herd_demand, jul_herd_demand, aug_herd_demand, sep_herd_demand, oct_herd_demand, nov_herd_demand, dec_herd_demand, total_herd_demand

#Update monthly grazing system demand
jan_system_demand = jan_system_demand + jan_herd_demand
feb_system_demand = feb_system_demand + feb_herd_demand
mar_system_demand = mar_system_demand + mar_herd_demand
apr_system_demand = apr_system_demand + apr_herd_demand
may_system_demand = may_system_demand + may_herd_demand
jun_system_demand = jun_system_demand + jun_herd_demand
jul_system_demand = jul_system_demand + jul_herd_demand
aug_system_demand = aug_system_demand + aug_herd_demand
sep_system_demand = sep_system_demand + sep_herd_demand
oct_system_demand = oct_system_demand + oct_herd_demand
nov_system_demand = nov_system_demand + nov_herd_demand
dec_system_demand = dec_system_demand + dec_herd_demand
total_system_demand = total_system_demand + total_herd_demand

#Send to output for this grazing system
If last herd in the grazing system
Output jan_system_demand, feb_system_demand, mar_system_demand, apr_system_demand, may_system_demand, jun_system_demand, jul_system_demand, aug_system_demand, sep_system_demand, oct_system_demand, nov_system_demand, dec_system_demand, total_system_demand

1.3. Output
grazing_system_year
herd_id … one or more
animal_group_id … one or more
animal_daily_intake_lbs
animal_monthly_intake_lbs
animal_yearly_intake_lbs
group_daily_intake_lbs
animal_aue
jan_group_demand
feb_group_demand
mar_group_demand
apr_group_demand
may_group_demand
jun_group_demand
jul_group_demand
aug_group_demand
sep_group_demand
oct_group_demand
nov_group_demand
dec_group_demand
total_group_demand
jan_herd_demand
feb_herd_demand

mar_herd_demand
apr_herd_demand
may_herd_demand
jun_herd_demand
jul_herd_demand
aug_herd_demand
sep_herd_demand
oct_herd_demand
nov_herd_demand
dec_herd_demand
total_herd_demand
jan_system_demand
feb_system_demand
mar_system_demand
apr_system_demand
may_system_demand
jun_system_demand
jul_system_demand
aug_system_demand
sep_system_demand
oct_system_demand
nov_system_demand
dec_system_demand
total_system_demand

Service GRAS-10: Calculate Daily and Cumulative Forage Supplies for the Grazing System (CalcForageSupply)

Purpose: Calculate forage supplies for each area of analysis (AoA) in the grazing system, accounting for forage partition profiles (FPPs).

Service Signature

Request Payload

grazing_system_start_date … e.g. January 1, 2016 (in suitable date format)
grazing_system_end_date… e.g. December 31, 2020 (in suitable date format)
AoA identifier … one or more in the grazing system
    starting_aoa_forage_prod … at beginning of first year of the grazing system (lbs/acres)
    aoa_acres … Area of Analysis acres

#Daily AoA forage production by month; includes FAA adjustment
AoA identifier … one or more in the grazing system
    aoa_prod_jan_daily
    aoa_prod_feb_daily
    aoa_prod_feb_leap_daily
    aoa_prod_mar_daily
    aoa_prod_apr_daily
    aoa_prod_may_daily
    aoa_prod_jun_daily
    aoa_prod_jul_daily
    aoa_prod_aug_daily
    aoa_prod_sep_daily
    aoa_prod_oct_daily
    aoa_prod_nov_daily
    aoa_prod_dec_daily

#AoA Forage partition profiles (FPPs) for each year in the grazing system
AoA identifier … one or more in the grazing system
    land_use
    fpp_activity_id … one or more for each year in the grazing system per AoA
        fpp_activity_type … values are restricted use, harvest roughage, always available
        harvest_efficiency_pct
        calendar_start_day … e.g. March 15, 2017 (in suitable date format)
        calendar_end_day … e.g. May 17, 2017 (in suitable date format)

Result Payload

system_forage_supply … total for entire grazing system
grazing_system_year … one or more

this_year_forage_supply
jan_forage_supply
feb_forage_supply
mar_forage_supply
apr_forage_supply
may_forage_supply
jun_forage_supply
jul_forage_supply
aug_forage_supply
sep_forage_supply
oct_forage_supply
nov_forage_supply
dec_forage_supply

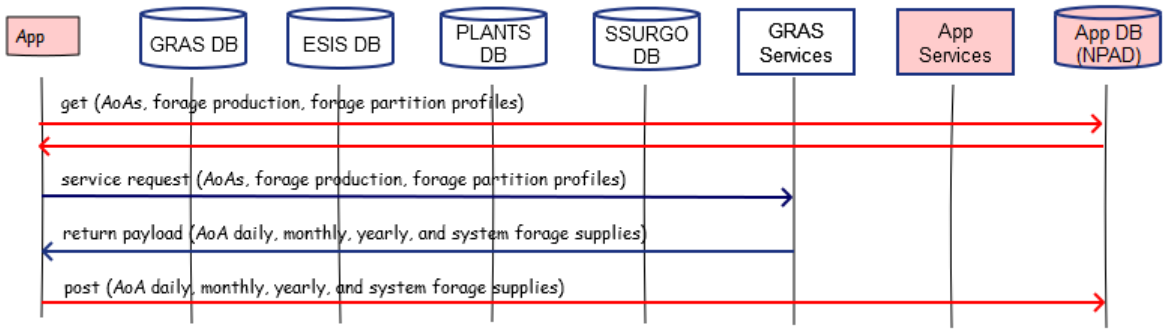
AoA identifier … one or more in the grazing system
grazing_system_date … each day through the years in the grazing system
this_day_aoa_forage_supply … will be NULL during restricted use and harvest roughage periods
cumulative_aoa_forage_supply
aoa_forage_harvest
grazing_system_year … one or more
jan_aoa_forage_supply
feb_aoa_forage_supply
mar_aoa_forage_supply
apr_aoa_forage_supply
may_aoa_forage_supply
jun_aoa_forage_supply
jul_aoa_forage_supply
aug_aoa_forage_supply
sep_aoa_forage_supply
oct_aoa_forage_supply
nov_aoa_forage_supply
dec_aoa_forage_supply
this_year_aoa_forage_supply

grazing_system_year … one or more
month … one or more
roughage_harvest_amt

Reference Data Sources

None accessed by this service

GRAS-10: Calculate Daily, Monthly, and Yearly, and System Forage Supplies



Component

1. **Calculate Daily, Monthly, and Cumulative Forage Supplies by AoA and System (CalcAoASysSupplies)**

   1.1. Inputs

   All from request payload

   1.2. Methods

   #Calculate daily forage supply, update cumulative totals, calculate monthly, annual, and system forage supply, and calculate monthly roughage harvested amount

   For each day (date) in the grazing system

   #Set this day, month of this day, and year of this day

   this_day = date in year of the grazing system

   previous_day = current date – 1 (one day earlier)

   If leap year (e.g. 2016, 2020, 2024, etc.)

   this_month = month of this_day in the leap year

   Else

   this_month = month of this_day in the non-leap year

   grazing_system_year = year in the grazing system (e.g. 2018)

   #Calculate forage supply for each AoA on this day

   For each AoA in the grazing system on this day

   this_aoa = AoA identifer

   #Calculate initial forage supply for this AoA on first day of grazing system

   starting_aoa_forage_supply = starting_aoa_forage_prod * aoa_acres

   #Set initial forage supply for this AoA on first day of grazing system

   If this_day == grazing_system_start_date

   cumulative_aoa_forage_supply = starting_aoa_forage_supply

   cumulative_forage_prod = starting_aoa_forage_supply

   aa_carryover_prod = 0

   restricted_use_period_prod = 0

   harvest_roughage_period_prod = 0

#Get forage partition profile (FPP) activity, harvest efficiency, and end date from request payload for this AoA on this day and the previous day
fpp_activity_today = fpp_activity_type of the fpp_activity_id where this_day >= calendar_start_date and <= calendar_end_date

fpp_activity_yesterday = fpp_activity_type of the fpp_activity_id where previous_day >= calendar_start_date and <= calendar_end_date

this_day_harvest_efficiency = harvest_efficiency_pct of the fpp_activity_id where this_day >= calendar_start_date and <= calendar_end_date

previous_day_harvest_efficiency = harvest_efficiency_pct of the fpp_activity_id where previous_day >= calendar_start_date and <= calendar_end_date

fpp_activity_end_date = calendar_end_day of the fpp_activity_id where this_day >= calendar_start_day and <= calendar_end_day

#Set daily production for this AoA on this day from request payload for the month of this day
If this_month == January
    this_day_aoa_prod = aoa_prod_jan_daily
Else if this_month == February and not leap year
    this_day_aoa_prod = aoa_prod_feb_daily
Else if this month == February and leap year
    this_day_aoa_prod = aoa_prod_feb_leap_daily
Else if this_month == March
    this_day_aoa_prod = aoa_prod_mar_daily
Else if this_month == April
    this_day_aoa_prod = aoa_prod_apr_daily
Else if this_month == May
    this_day_aoa_prod = aoa_prod_may_daily
Else if this_month == June
    this_day_aoa_prod = aoa_prod_jun_daily
Else if this_month == July
    this_day_aoa_prod = aoa_prod_jul_daily
Else if this_month == August
    this_day_aoa_prod = aoa_prod_aug_daily
Else if this_month == September
    this_day_aoa_prod = aoa_prod_sep_daily
Else if this_month == October
    this_day_aoa_prod = aoa_prod_oct_daily
Else if this_prod == November
    this_day_aoa_prod = aoa_prod_nov_daily
Else if this_month == December
    this_day_aoa_prod = aoa_prod_dec_daily

#Calculate cumulative forage production and supply and today's forage supply for this AoA on this day

```
If fpp_activity_today == restricted use
    # Calculate cumulative forage production for this day
    cumulative_forage_prod = cumulative_forage_prod + this_day_aoa_prod

    #No daily forage supply calculation during restricted use FPP
    this_day_aoa_forage_supply = 0

    If this_day == grazing_system_start_date
        cumulative_aoa_forage_supply = 0

    Else
        cumulative_aoa_forage_supply = cumulative_aoa_forage_supply +
        this_day_aoa_forage_supply

    #Update a restricted use period production value, which will be used for
    calculating the daily forage supply on the first day of the next FPP period
    (either harvest roughage or always available)
    If this_day == grazing_system_start_date
        restricted_use_period_prod = cumulative_forage_prod +
        this_day_aoa_prod
    Else
        restricted_use_period_prod = restricted_use_period_prod +
        this_day_aoa_prod

    #If end of restricted use FPP period on this day
    If this_day == fpp_activity_end_day

        #Reset always available carryover forage production to zero
        aa_carryover_prod = 0

Else if fpp_activity_today == harvest roughage
    # Calculate cumulative forage production for this day
        cumulative_forage_prod = cumulative_forage_prod +
        this_day_aoa_prod

    #No daily forage supply calculation during harvest roughage FPP
    this_day_aoa_forage_supply = 0

    If this_day == grazing_system_start_date
        cumulative_aoa_forage_supply = 0

    Else
        cumulative_aoa_forage_supply = cumulative_aoa_forage_supply +
        this_day_aoa_forage_supply
```

```
#Calculate a harvest roughage period production value
If this_day == grazing_system_start_date
    harvest_roughage_period_prod = cumulative_forage_prod +
    this_day_aoa_prod

Else
    harvest_roughage_period_prod = harvest_roughage_period_prod +
    this_day_aoa_prod

#If end of harvest roughage FPP period on this day
If this_day == fpp_activity_end_day

    #Harvest roughage from this AoA (cut for hay, to the barn)
    aoa_forage_harvest = (harvest_roughage_period_prod +
    aa_carryover_prod + restricted_use_period_prod) *
    this_day_harvest_efficiency

    #Adjust harvest roughage period production remaining on the AoA
    harvest_roughage_period_prod = (harvest_roughage_period_prod +
    aa_carryover_prod + restricted_use_period_prod) –
    aoa_forage_harvest

    #Reset always available carryover forage production to zero
    aa_carryover_prod = 0

    #Reset restricted_use_period_production to zero
    restricted_use_period_prod = 0

    #Output roughage harvested for the grazing system year and month
    of this date
    Add AoA, grazing_system_year, this_month, aoa_forage_harvest to
    output array
    If any array rows have same year and month
        combine into one row
            Aoa, grazing_system_year, this_month,
            sum(aoa_forage_harvest)


Else if fpp_activity_today == always available
    #Calculate daily AoA forage supply by applying harvest efficiency; when
    specified include stockpiled or harvested forage
    If fpp_activity_yesterday == restricted use
        stockpiled_aoa_forage = restricted_use_period_prod * (1-
        previous_day_harvest_efficiency)
        this_day_aoa_forage_supply = (stockpiled_aoa_forage +
        this_day_aoa_prod) * this_day_harvest_efficiency
```

```
        #Reset restricted_use_period_production to zero
        restricted_use_period_prod = 0

    Else If fpp_activity_yesterday == harvest roughage
        this_day_aoa_forage_supply = (harvested_roughage_period_prod +
        this_day_aoa_prod) * this_day_harvest_efficiency


        #Reset harvested roughage period production to zero
        harvested_roughage_period_prod = 0

        cumulative_forage_prod = (cumulative_forage_prod +
        this_day_aoa_prod) – aoa_forage_harvest

    Else
        this_day_aoa_forage_supply = this_day_aoa_prod *
        this_day_harvest_efficiency

        cumulative_forage_prod = cumulative_forage_prod +
        this_day_aoa_prod

    #Calculate cumulative AoA forage supply
    If this_day == grazing_system_start_date
        cumulative_aoa_forage_supply = (cumulative_aoa_forage_supply *
        this_day_harvest_efficiency) + this_day_aoa_forage_supply

    Else
        cumulative_aoa_forage_supply = cumulative_aoa_forage_supply +
        this_day_aoa_forage_supply


         #If end of always available FPP period on this day calculate carryover
         forage production
         If this_day == fpp_activity_end_date
             aa_carryover_prod = cumulative_forage_prod –
             cumulative_aoa_forage_supply

    #Update cumulative forage supply to this day all AoAs
    cumulative_forage_supply = cumulative_forage_supply +
    cumulative_aoa_forage_supply

#Send daily and cumulative forage supply to Output
Output daily and cumulative forage supply for this AoA on this day as JSON

#Set initial forage supply for this AoA on first day of grazing system
If this_day == grazing_system_start_date
    prev_aoa_forage_supply = 0
```

prev_year_aoa_forage_supply = 0
prev_year_forage_supply = 0

#As applicable calculate AoA month supply and update month supply across all AoAs for this year in grazing system

If this_day == January 31st
jan_aoa_forage_supply = cumulative_aoa_forage_supply – prev_aoa_forage_supply
jan_forage_supply = jan_forage_supply + jan_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if leap year and this_day == February 29st
feb_aoa_forage_supply = cumulative_aoa_forage_supply – prev_aoa_forage_supply
feb_forage_supply = feb_forage_supply + feb_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == February 28st
feb_aoa_forage_supply = cumulative_aoa_forage_supply – prev_aoa_forage_supply
feb_forage_supply = feb_forage_supply + feb_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == March 31st
mar_aoa_forage_supply = cumulative_aoa_forage_supply – mar_forage_supply = mar_forage_supply + mar_aoa_forage_supply
prev_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == April 30th
apr_aoa_forage_supply = cumulative_aoa_forage_supply – prev_aoa_forage_supply
apr_forage_supply = apr_forage_supply + apr_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == May 31st
may_aoa_forage_supply = cumulative_aoa_forage_supply – prev_aoa_forage_supply
may_forage_supply = may_forage_supply + may_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == June 30th
jun_aoa_forage_supply = cumulative_aoa_forage_supply – prev_aoa_forage_supply
jun_forage_supply = jun_forage_supply + jun_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == July 31st
jul_aoa_forage_supply = cumulative_aoa_forage_supply – prev_aoa_forage_supply
jul_forage_supply = jul_forage_supply + jul_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == August 31st
aug_aoa_forage_supply = cumulative_aoa_forage_supply –

prev_aoa_forage_supply
aug_forage_supply = aug_forage_supply + aug_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == September 30th
sep_aoa_forage_supply = cumulative_aoa_forage_supply –
prev_aoa_forage_supply
sep_forage_supply = sep_forage_supply + sep_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == October 31st
oct_aoa_forage_supply = cumulative_aoa_forage_supply –
prev_aoa_forage_supply
oct_forage_supply = oct_forage_supply + oct_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == November 30th
nov_aoa_forage_supply = cumulative_aoa_forage_supply –
prev_aoa_forage_supply
nov_forage_supply = nov_forage_supply + nov_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply
Else if this_day == December 31st (last day of this year)
dec_aoa_forage_supply = cumulative_aoa_forage_supply –
prev_aoa_forage_supply
dec_forage_supply = dec_forage_supply + dec_aoa_forage_supply
prev_aoa_forage_supply = cumulative_aoa_forage_supply

#Calculate total forage supply for this AoA for this year
this_year_aoa_forage_supply = cumulative_aoa_forage_supply –
prev_year_aoa_forage_supply
prev_year_aoa_forage_supply = cumulative_aoa_forage_supply

#Send monthly and this year AoA forage supply to Output
For this_year and this_aoa
output jan_aoa_forage_supply, feb_aoa_forage_supply,
mar_aoa_forage_supply, apr_aoa_forage_supply,
may_aoa_forage_supply, jun_aoa_forage_supply, jul_aoa_forage_supply,
aug_aoa_forage_supply, sep_aoa_forage_supply, oct_aoa_forage_supply,
nov_aoa_forage_supply, dec_aoa_forage_supply,
this_year_aoa_forage_supply

#Persist and pass monthly AoA forage supply data to next day iiteration
this_aoa, jan_aoa_forage_supply, feb_aoa_forage_supply,
mar_aoa_forage_supply, apr_aoa_forage_supply, may_aoa_forage_supply,
jun_aoa_forage_supply, jul_aoa_forage_supply, aug_aoa_forage_supply,
sep_aoa_forage_supply, oct_aoa_forage_supply, nov_aoa_forage_supply,
dec_aoa_forage_supply, this_year_aoa_forage_supply,

#Persist and pass monthly forage supply data to next day iiteration
jan_forage_supply, feb_forage_supply, mar_forage_supply,

apr_forage_supply, may_forage_supply, jun_forage_supply,
jul_forage_supply, aug_forage_supply, sep_forage_supply,
oct_forage_supply, nov_forage_supply, dec_forage_supply

If this_day == December 31st (last day of this year)
this_year_forage_supply = cumulative_forage_supply –
prev_year_forage_supply
prev_year_forage_supply = cumulative_forage_supply

#Send to Output
Output grazing_system_year, this_year_forage_supply, jan_forage_supply,
feb_forage_supply, mar_forage_supply, apr_forage_supply,
may_forage_supply, jun_forage_supply, jul_forage_supply,
aug_forage_supply, sep_forage_supply, oct_forage_supply,
nov_forage_supply, dec_forage_supply

If this_day == grazing_system_end_date
system_forage_supply = cumulative_forage_supply

#Send to Output
Output this_year, this_year_forage_supply

#Calculate Monthly Harvested Roughage Amount
For each grazing_system_year
For each month
roughage_harvest_amt = roughage_harvest_amt + aoa_forage_harvest

1.3. Output
system_forage_supply … total for entire grazing system
grazing_system_year … one or more
this_year_forage_supply
jan_forage_supply
feb_forage_supply
mar_forage_supply
apr_forage_supply
may_forage_supply
jun_forage_supply
jul_forage_supply
aug_forage_supply
sep_forage_supply
oct_forage_supply
nov_forage_supply
dec_forage_supply

AoA identifier … one or more in the grazing system
grazing_system_date … each day through the years in the grazing system

this_day_aoa_forage_supply … will be NULL during restricted use and harvest roughage periods
cumulative_aoa_forage_supply
grazing_system_year … one or more
jan_aoa_forage_supply
feb_aoa_forage_supply
mar_aoa_forage_supply
apr_aoa_forage_supply
may_aoa_forage_supply
jun_aoa_forage_supply
jul_aoa_forage_supply
aug_aoa_forage_supply
sep_aoa_forage_supply
oct_aoa_forage_supply
nov_aoa_forage_supply
dec_aoa_forage_supply
this_year_aoa_forage_supply

grazing_system_year … one or more
month … one or more
roughage_harvest_amt

Service GRAS-11: Calculate Grazing Schedule Period Forage Animal Balance (CalcPeriodFAB)

Purpose: Calculate a forage animal balance for a period as one or more herds are assigned to graze available forage on a grazing unit in the grazing system. A grazing unit corresponds to an area of analysis (AoA) and almost always to a NRCS planning land unit (PLU); they are interchangeable terms. The service provides a daily accounting of forage animal balance, forage supply, and animal demand during each grazing period of the AoAs of the grazing schedule.

The application user begins by opening a grazing schedule dialog. The application retreives saved herd and animal group, forage supply, and grazing schedule data for the grazing system and calls this service.

The GRAS grazing schedule will present data similar to the following. The results payload of this service will provide data to populate this dialog. The user schedules herds to PLUs (AoAs) and the application calls the service to update the FAB for each AoA and grazing system. The service also calculates a grazing period FAB upon which the color coding is based.

PRESCRIBED GRAZING SCHEDULE

(grouped by herd)

Year:	2015

Herd	PLU	Total Acres	Total Grazeable Forage Lbs	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	Forage Balance
1	1	200	350,000													400
1	2	300	243,676													600
2	2	300	243,676													600
3	1	200	350,000													400
4	3	250	345,678	Prescribed Burn												-1609
Total		750												Total Balance		-609

PRESCRIBED GRAZING SCHEDULE

(grouped by PLU)

Year:	2015

Herd	PLU	Total Acres	Total Grazeable Forage Lbs	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	Forage Balance
1	1	200	350,000													400
3																
1	2	300	243,676													600
2																
4	3	250	345,678	Prescribed Burn												-1609
Total		750												Total Balance		-609

Service Signature

Request Payload

grazing_system_start_date … e.g. January 1, 2016 (in suitable date type); Grazing System Start Date
grazing_system_end_date… e.g. December 31, 2020 (in suitable date type); Grazing System End Date
AoAId … one or more in the grazing system; Area of Analysis Identifier;
grazing_system_date … date type; Day in the Grazing System;
this_day_aoa_forage_supply … integer; Forage Supply on This Day; units: pounds/acre

#Input data for calculating daily animal demand; should be sufficient number of herds and groups for the grazing expected during the entire period of the grazing system, mediated by date in and out of herd.
herd_id … one or more in the grazing system
animal_group_id … one or more in the herd
animal_unit_id
number_of_head
average_weight_lbs
avg_daily_intake_pct
date_into_herd
date_out_of_herd

#Input data for herd assignment to AoAs during always available FPP activity periods
AoA identifier … one or more in the grazing system
graze_period_id … one or more on the AoA
graze_period_start_date … earliest herd graze start date
graze_period_end_date… latest herd graze end date
herd_id … one or more in the grazing period
graze_start_date … must be within always available FPP activity
graze_end_date … must be within always available FPP activity

Result Payload
#Grazing system forage animal balance for all AoAs
System_fab

#Grazing system forage animal balance for each year for all AoAs
grazing_system_year
system_annual_fab

AoA identifier … one or more in the grazing system
grazing_period_id … zero to many in the AoA spanning years
 # Forage animal balance, forage supply, and animal demand for each day in the scheduled grazing period for each AoA
graze_period_date
this_day_aoa_fab
this_day_aoa_forage_supply
this_day_aoa_animal_demand

 # Cumulative forage animal balance, forage supply, and animal demand for the scheduled grazing period for each AoA, and days the balance was negative
period_aoa_fab
period_aoa_forage_supply
period_aoa_animal_demand
period_fab_negative_days

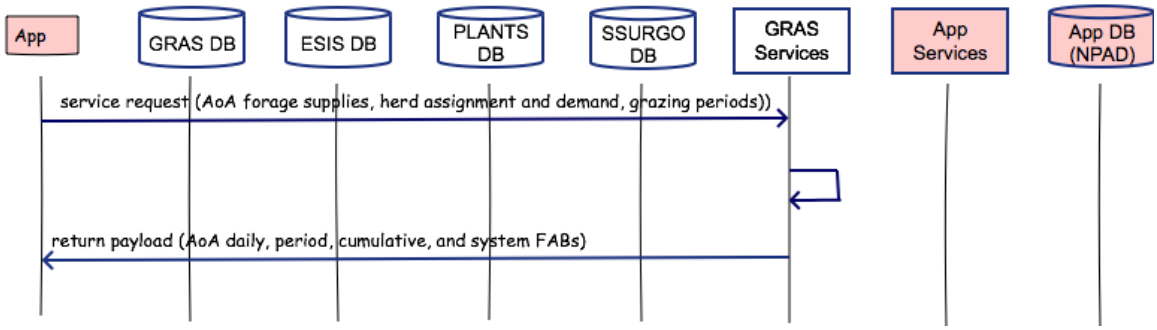
#Annual and grazing system forage animal balance for the AoA
grazing_system_year

cum_aoa_fab
system_aoa_fab

Reference Data Sources

None accessed by this service

GRAS-11: Calculate Grazing Schedule Period Forage Animal Balances



Component

1. **Calculate Grazing Period Animal Demand and Forage Animal Balance (CalcPeriodd)**

1.1. Inputs

All inputs in request payload

1.2. Methods

#Calculate grazing period demand for the AoAs in the grazing system
For each day (date) in the grazing system
#Set this day, month of this day, and year of this day
this_day = date in year of the grazing system
If leap year (e.g. 2016, 2020, 2024, etc.)
this_month = month of this_day in the leap year
Else
this_month = month of this_day in the non-leap year
grazing_system_year = year in the grazing system (e.g. 2018)

#Calculate forage supply, animal demand, and forage animal balance (FAB) for each AoA on this day
For each AoA in the grazing system on this day
this_aoa = AoA identifier

#Set initial animal demand for this AoA on first day of grazing system
If this_day == grazing_system_start_date
this_day_aoa_animal_demand = 0.00

For each graze_period_id in AoA
#Calculate animal demand for this AoA on this day
For each herd_id in the graze period

If this_day >= graze_start_date and <= graze_end_date
For each animal_group_id in the herd
this_day_aoa_animal_demand =
this_day_aoa_animal_demand + (number_of_head *
average_weight_lbs * avg_daily_intake_pct)

#Calculate forage animal balance on this day for this AoA
For this AoA and grazing_system_date == this_day
this_day_aoa_fab = this_day_aoa_forage_supply –
this_day_aoa_animal_demand

graze_period_date = this_day

#Send this day's forage animal balance, forage supply, and animal demand to Output for this AoA
Output AoA identifier, graze_period_id, graze_period_date,
this_day_aoa_fab, this_day_aoa_forage_supply,
is_day_aoa_animal_demand

#Count negative FAB days in grazing period for this AoA
If this_day_aoa_fab < 0.00
period_fab_negative_days = fab_negative_days + 1

#Update cumulative grazing period FAB, forage supply, and animal demand for this AoA
If this_day >= graze_period_start_date <= graze_period_end_date
period_aoa_fab = period_aoa_fab + this_day_aoa_fab

period_aoa_forage_supply = period_aoa_forage_supply +
this_day_aoa_forage_supply

period_aoa_animal_demand = period_aoa_animal_demand +
this_day_aoa_animal_demand

#Persist and pass this AoA grazing period data to next day iteration
If this_day < graze_period_end_date
Pass to next day: graze_period_id, period_aoa_fab,
period_aoa_forage_supply, period_aoa_animal_demand,
period_fab_negative_days

#Send period forage animal balance, forage supply, and animal demand to Output
If this_day == graze_period_end_date
Output Aoa identifier, graze_period_id, period_aoa_fab,
period_aoa_forage_supply, period_aoa_animal_demand,
period_fab_negative_days

#Update cumulative AoA forage animal balance for this year
cum_aoa_fab = cum_aoa_fab + this_day_aoa_fab

#Send cumulative AoA forage animal balance to Output at end of year
If this_day == December 31st of this grazing system year
Output AoA identifier, grazing_system_year, cum_aoa_fab

Add AoA identifier, grazing_system_year, cum_aoa_fab to a fab_annual array

#Reset cumulative AoA forage animal balance to zero for next year
cum_aoa_fab = 0.00**#Update cumulative system AoA forage animal balance**
system_aoa_fab = system_aoa_fab + this_day_aoa_fab
If this_day == grazing_system_end_date
Output Aoa identifier, system_aoa_fab

Add AoA identifier, system_aoa_fab to a fab_system array

#Calculate system annual FAB
If last AoA in grazing system sum cum_aoa_fab for fab_annual array rows with the same grazing system year
For each grazing_system_year
system_annual_fab = sum(cum_aoa_fab)

Output grazing_system_year, system_annual_fab

#Calculate system FAB
If last AoA in grazing system sum system_aoa_fab for all fab_system array rows
system_fab = sum(system_aoa_fab)

Output system_fab

1.3. Output
system_fab

grazing_system_year
system_annual_fab

AoA identifier … one or more in the grazing system
grazing_period_id … zero to many in the AoA spanning years
graze_period_date
this_day_aoa_fab
this_day_aoa_forage_supply
this_day_aoa_animal_demand
period_aoa_fab
period_aoa_forage_supply

period_aoa_animal_demand
period_fab_negative_days
grazing_system_year
cum_aoa_fab
system_aoa_fab

2. <u>Calculate Grazing System Year Forage Animal Balance (CalcSystemFAB)</u>

2.1. Inputs

 # From previous component of this service
grazing_system_year … one or more in the grazing system
cum_aoa_fab

2.2. Methods

#Calculate grazing system year forage animal balance for all the AoAs in the grazing system
For each grazing system year
For each AoA
system_year_fab = system_year_fab + cum_aoa_fab

If last AoA in the grazing system for this year
#Send grazing system year forage animal balance for all AoAs to Output
Output grazing_system_year, system_year_fab

2.3. Output

grazing_system_year
system_year_fab

3. <u>Calculate Total System Forage Animal Balance (CalcSystemAB)</u>

3.1. Inputs

#From previous component of this service
grazing_system_year … one or more in the grazing system
system_year_fab

3.2. Methods

#Calculate total system forage animal balance for all years in the grazing system
For each grazing system year
system_fab = system_fab + system_year_fab

If last year in the grazing system
#Send system forage animal balance for all years to Output
Output system_fab

3.3. Output

system_fab

Service GRAS-12: Calculate Monthly, Yearly, and System Forage Animal Balance Without Grazing Schedule (CalcBasicFAB)

Purpose: Calculate monthly, yearly, and system forage animal balances for a grazing system without a grazing schedule. Forage supply and animal demand are calculated daily through all years in the grazing system, accounting for harvested roughage, forage partition profile activities and harvest efficiencies.

Service Signature

Request Payload

grazing_system_start_date … e.g. January 1, 2016 (in suitable date format)
grazing_system_end_date… e.g. December 31, 2020 (in suitable date format)

#Annual total and monthly forage supplies for the grazing system
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
this_year_forage_supply … numeric (10,1); This Year Forage Supply
jan_forage_supply … numeric (9,1); January Forage Supply
feb_forage_supply … numeric (9,1); February Forage Supply
mar_forage_supply … numeric (9,1); March Forage Supply
apr_forage_supply … numeric (9,1); April Forage Supply
may_forage_supply … numeric (9,1); May Forage Supply
jun_forage_supply … numeric (9,1); June Forage Supply
jul_forage_supply … numeric (9,1); July Forage Supply
aug_forage_supply … numeric (9,1); August Forage Supply
sep_forage_supply … numeric (9,1); September Forage Supply
oct_forage_supply … numeric (9,1); October Forage Supply
nov_forage_supply … numeric (9,1); November Forage Supply
dec_forage_supply … numeric (9,1); December Forage Supply

#AoA monthly forage supply for each year in the grazing system
AoAId … integer; one or more in the grazing system; Area of Analysis Identifier
aoa_acres … decimal (10,2); Area of Analysis Acres
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
jan_aoa_forage_supply … numeric (9,1); January AoA Forage Supply
feb_aoa_forage_supply … numeric (9,1); February AoA Forage Supply
mar_aoa_forage_supply … numeric (9,1); March AoA Forage Supply
apr_aoa_forage_supply … numeric (9,1); April AoA Forage Supply
may_aoa_forage_supply … numeric (9,1); May AoA Forage Supply
jun_aoa_forage_supply … numeric (9,1); June AoA Forage Supply
jul_aoa_forage_supply … numeric (9,1); July AoA Forage Supply
aug_aoa_forage_supply … numeric (9,1); August AoA Forage Supply
sep_aoa_forage_supply … numeric (9,1); September AoA Forage Supply
oct_aoa_forage_supply … numeric (9,1); October AoA Forage Supply

nov_aoa_forage_supply … numeric (9,1); November AoA Forage Supply
dec_aoa_forage_supply … numeric (9,1); December AoA Forage Supply
this_year_aoa_forage_supply … numeric (10,1); This Year AoA Forage Supply

#Amount of roughage fed on a monthly basis each year.
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
jan_roughage_supply … numeric (7,0); January Rouhage Supply
feb_roughage_supply … numeric (7,0); February Rouhage Supply
mar_roughage_supply … numeric (7,0); March Rouhage Supply
apr_roughage_supply … numeric (7,0); April Rouhage Supply
may_roughage_supply … numeric (7,0); May Rouhage Supply
jun_roughage_supply … numeric (7,0); June Rouhage Supply
jul_roughage_supply … numeric (7,0); July Rouhage Supply
aug_roughage_supply … numeric (7,0); August Rouhage Supply
sep_roughage_supply … numeric (7,0); September Rouhage Supply
oct_roughage_supply … numeric (7,0); October Rouhage Supply
nov_roughage_supply … numeric (7,0); November Rouhage Supply
dec_roughage_supply … numeric (7,0); December Rouhage Supply

#Feeding waste percentage that is utilized to adjust amount of roughage fed eachmonthly basis each year.
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
feeding_waste_pct … numeric (3,1); Percent of Feeding Waste

#Monthly herd demand and monthly system demand for each year in the grazing system
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
herd_id … smallint; one or more; Herd Identifier
jan_herd_demand … numeric (9,1); January Herd Demand
feb_herd_demand … numeric (9,1); February Herd Demand
mar_herd_demand … numeric (9,1); March Herd Demand
apr_herd_demand … numeric (9,1); April Herd Demand
may_herd_demand … numeric (9,1); May Herd Demand
jun_herd_demand … numeric (9,1); June Herd Demand
jul_herd_demand … numeric (9,1); July Herd Demand
aug_herd_demand … numeric (9,1); August Herd Demand
sep_herd_demand … numeric (9,1); September Herd Demand
oct_herd_demand … numeric (9,1); October Herd Demand
nov_herd_demand … numeric (9,1); November Herd Demand
dec_herd_demand … numeric (9,1); December Herd Demand
total_herd_demand … numeric (10,1); Total Herd Demand

jan_system_demand … numeric (9,1); January System Demand

feb_system_demand … numeric (9,1); February System Demand
mar_system_demand … numeric (9,1); March System Demand
apr_system_demand … numeric (9,1); April System Demand
may_system_demand … numeric (9,1); May System Demand
jun_system_demand … numeric (9,1); June System Demand
jul_system_demand … numeric (9,1); July System Demand
aug_system_demand … numeric (9,1); August System Demand
sep_system_demand … numeric (9,1); September System Demand
oct_system_demand … numeric (9,1); October System Demand
nov_system_demand … numeric (9,1); November System Demand
dec_system_demand … numeric (9,1); December System Demand
total_system_demand … numeric (10,1); Total System Demand

Result Payload

system_forage_animal_balance … numeric (10,1); Grazing System Forage Animal Balance
system_forage_roughage_supply … numeric (10,1); Grazing System Forage Supply including Roughage Fed
system_animal_demand … numeric (10,1); Grazing System Animal Demand
grazing_system_year … … smallint (e.g. 2016); one or more in the grazing system: Year in the Grazing System
AoAId … integer; one or more in the grazing system; Area of Analysis Identifier
aoa_acres … decimal (10,2); Area of Analysis Acres
jan_aoa_forage_supply … numeric (9,1); January AoA Forage Supply
feb_aoa_forage_supply … numeric (9,1); February AoA Forage Supply
mar_aoa_forage_supply … numeric (9,1); March AoA Forage Supply
apr_aoa_forage_supply … numeric (9,1); April AoA Forage Supply
may_aoa_forage_supply … numeric (9,1); May AoA Forage Supply
jun_aoa_forage_supply … numeric (9,1); June AoA Forage Supply
jul_aoa_forage_supply … numeric (9,1); July AoA Forage Supply
aug_aoa_forage_supply … numeric (9,1); August AoA Forage Supply
sep_aoa_forage_supply … numeric (9,1); September AoA Forage Supply
oct_aoa_forage_supply … numeric (9,1); October AoA Forage Supply
nov_aoa_forage_supply … numeric (9,1); November AoA Forage Supply
dec_aoa_forage_supply … numeric (9,1); December AoA Forage Supply
this_year_aoa_forage_supply … numeric (10,1); This Year AoA Forage Supply

jan_forage_supply … numeric (9,1); January Forage Supply
feb_forage_supply … numeric (9,1); February Forage Supply
mar_forage_supply … numeric (9,1); March Forage Supply
apr_forage_supply … numeric (9,1); April Forage Supply
may_forage_supply … numeric (9,1); May Forage Supply
jun_forage_supply… numeric (9,1); June Forage Supply
jul_forage_supply … numeric (9,1); July Forage Supply
aug_forage_supply … numeric (9,1); August Forage Supply

sep_forage_supply … numeric (9,1); September Forage Supply
oct_forage_supply … numeric (9,1); October Forage Supply
nov_forage_supply … numeric (9,1); November Forage Supply
dec_forage_supply … numeric (9,1); December Forage Supply
this_year_forage_supply… numeric (10,1); This Year Forage Supply

jan_roughage_fed … numeric (8,1); January Rouhage Fed
feb_roughage_fed … numeric (8,1); February Rouhage Fed
mar_roughage_fed … numeric (8,1); March Rouhage Fed
apr_roughage_fed … numeric (8,1); April Rouhage Fed
may_roughage_fed … numeric (8,1); May Rouhage Fed
jun_roughage_fed … numeric (8,1); June Rouhage Fed
jul_roughage_fed … numeric (8,1); July Rouhage Fed
aug_roughage_fed … numeric (8,1); August Rouhage Fed
sep_roughage_fed … numeric (8,1); September Rouhage Fed
oct_roughage_fed … numeric (8,1); October Rouhage Fed
nov_roughage_fed … numeric (8,1); November Rouhage Fed
dec_roughage_fed … numeric (8,1); December Rouhage Fed
this_year_roughage_fed … numeric (9,1); This Year Rouhage Fed

jan_forage_roughage_supply … numeric (10,1); January Forage Supply Including Rouhage Fed
feb_forage_roughage_supply … numeric (10,1); February Forage Supply Including Rouhage Fed
mar_forage_roughage_supply … numeric (10,1); March Forage Supply Including Rouhage Fed
apr_forage_roughage_supply … numeric (10,1); April Forage Supply Including Rouhage Fed
may_forage_roughage_supply … numeric (10,1); May Forage Supply Including Rouhage Fed
jun_forage_roughage_supply … numeric (10,1); June Forage Supply Including Rouhage Fed
jul_forage_roughage_supply … numeric (10,1); July Forage Supply Including Rouhage Fed
aug_forage_roughage_supply … numeric (10,1); August Forage Supply Including Rouhage Fed
sep_forage_roughage_supply … numeric (10,1); September Forage Supply Including Rouhage Fed
oct_forage_roughage_supply … numeric (10,1); October Forage Supply Including Rouhage Fed
nov_forage_roughage_supply … numeric (10,1); November Forage Supply Including Rouhage Fed
dec_forage_roughage_supply … numeric (10,1); December Forage Supply Including Rouhage Fed
this_year_forage_roughage_supply… numeric (10,1); This Year Forage Supply Including Rouhage Fed

herd_id … smallint; one or more; Herd Identifier
jan_herd_demand … numeric (9,1); January Herd Demand
feb_herd_demand … numeric (9,1); February Herd Demand
mar_herd_demand … numeric (9,1); March Herd Demand
apr_herd_demand … numeric (9,1); April Herd Demand
may_herd_demand … numeric (9,1); May Herd Demand
jun_herd_demand … numeric (9,1); June Herd Demand
jul_herd_demand … numeric (9,1); July Herd Demand
aug_herd_demand … numeric (9,1); August Herd Demand
sep_herd_demand … numeric (9,1); September Herd Demand
oct_herd_demand … numeric (9,1); October Herd Demand
nov_herd_demand … numeric (9,1); November Herd Demand
dec_herd_demand … numeric (9,1); December Herd Demand
this_year_herd_demand … numeric (10,1); This Year Herd Demand

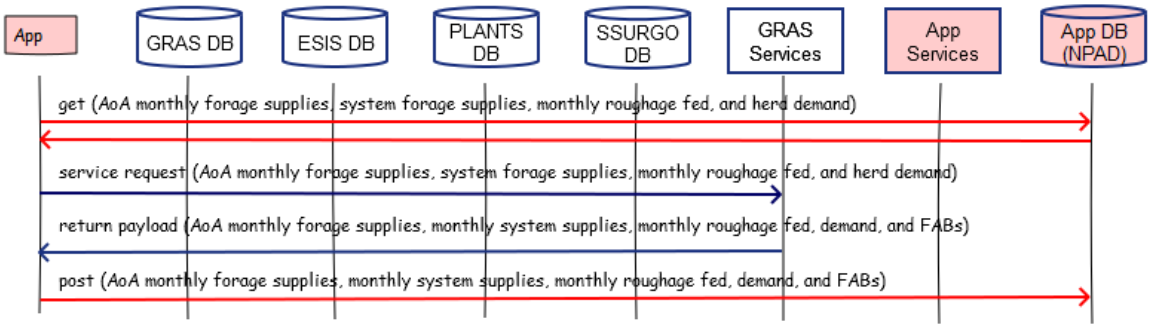
jan_system_demand … numeric (9,1); January System Demand
feb_system_demand … numeric (9,1); February System Demand
mar_system_demand … numeric (9,1); March System Demand
apr_system_demand … numeric (9,1); April System Demand
may_system_demand … numeric (9,1); May System Demand
jun_system_demand … numeric (9,1); June System Demand
jul_system_demand … numeric (9,1); July System Demand
aug_system_demand … numeric (9,1); August System Demand
sep_system_demand … numeric (9,1); September System Demand
oct_system_demand … numeric (9,1); October System Demand
nov_system_demand … numeric (9,1); November System Demand
dec_system_demand … numeric (9,1); December System Demand
total_system_demand … numeric (10,1); Total System Demand

fab_january … numeric (9,1); January Forage Animal Balance
fab_february … numeric (9,1); February Forage Animal Balance
fab_march … numeric (9,1); March Forage Animal Balance
fab_april … numeric (9,1); April Forage Animal Balance
fab_may … numeric (9,1); May Forage Animal Balance
fab_june … numeric (9,1); June Forage Animal Balance
fab_july … numeric (9,1); July Forage Animal Balance
fab_august … numeric (9,1); August Forage Animal Balance
fab_september … numeric (9,1); September Forage Animal Balance
fab_october … numeric (9,1); October Forage Animal Balance
fab_november … numeric (9,1); November Forage Animal Balance
fab_december … numeric (9,1); December Forage Animal Balance
this_year_forage_animal_balance … numeric (10,1); This Year Forage Animal Balance

Reference Data Sources

none

GRAS-12: Calculate Forage Animal Balance Without Grazing Schedule



Component

1. Calculate Basic Forage Animal Balance (CalcBasicFAB)

1.1. Inputs

#From request payload

grazing_system_start_date … e.g. January 1, 2016 (in suitable date format)
grazing_system_end_date… e.g. December 31, 2020 (in suitable date format)

AoAId … one or more in the grazing system
- aoa_acres
- grazing_system_year … one or more in the grazing system
  - jan_aoa_forage_supply
  - feb_aoa_forage_supply
  - mar_aoa_forage_supply
  - apr_aoa_forage_supply
  - may_aoa_forage_supply
  - jun_aoa_forage_supply
  - jul_aoa_forage_supply
  - aug_aoa_forage_supply
  - sep_aoa_forage_supply
  - oct_aoa_forage_supply
  - nov_aoa_forage_supply
  - dec_aoa_forage_supply
  - this_year_aoa_forage_supply

grazing_system_year … one or more in the grazing system

#From request payload

- jan_forage_supply
- feb_forage_supply
- mar_forage_supply
- apr_forage_supply
- may_forage_supply
- jun_forage_supply
- jul_forage_supply
- aug_forage_supply

sep_forage_supply
oct_forage_supply
nov_forage_supply
dec_forage_supply
this_year_forage_supply

#From request payload
grazing_system_year … one or more in the grazing system
jan_roughage_supply
feb_roughage_supply
mar_roughage_supply
apr_roughage_supply
may_roughage_supply
jun_roughage_supply
jul_roughage_supply
aug_roughage_supply
sep_roughage_supply
oct_roughage_supply
nov_roughage_supply
dec_roughage_supply

#From request payload
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
feeding_waste_pct

#Monthly herd demand and monthly system demand for each year in the grazing system
grazing_system_year
herd_id
jan_herd_demand
feb_herd_demand
mar_herd_demand
apr_herd_demand
may_herd_demand
jun_herd_demand
jul_herd_demand
aug_herd_demand
sep_herd_demand
oct_herd_demand
nov_herd_demand
dec_herd_demand
total_herd_demand

jan_system_demand
feb_system_demand
mar_system_demand

apr_system_demand
may_system_demand
jun_system_demand
jul_system_demand
aug_system_demand
sep_system_demand
oct_system_demand
nov_system_demand
dec_system_demand
total_system_demand

1.2. Methods

For each year in the grazing system
grazing_system_year = year in the grazing system (e.g. 2018)

#Calculate monthly roughage added for each year

For each year in the grazing system
If year >= grazing_system_start_date and <=grazing_system_end_date
For each month in the year
If month == January
jan_roughage_fed = jan_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed +
jan_roughage_fed
Else if month == February
feb_roughage_fed = feb_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed +
feb_roughage_fed
Else if month == March
mar_roughage_fed = mar_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed +
mar_roughage_fed
Else if month == April
apr_roughage_fed = apr_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed +
apr_roughage_fed
Else if month == May
may_roughage_fed = may_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed +
may_roughage_fed
Else if month == June
jun_roughage_fed = jun_roughage_supplys * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed +
jun_roughage_fed
Else if month == July
jul_roughage_fed = jul_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed +
jul_roughage_fed

```
Else if month == August
    aug_roughage_fed = aug_roughage_supply * (1 – feeding_waste_pct)
    cumulative_roughage_fed = cumulative_roughage_fed +
    aug_roughage_fed
Else if month == September
    sep_roughage_fed = sep_roughage_supply * (1 – feeding_waste_pct)
    cumulative_roughage_fed = cumulative_roughage_fed +
    sep_roughage_fed
Else if month == October
    oct_roughage_fed = oct_roughage_supply* (1 – feeding_waste_pct)
    cumulative_roughage_fed = cumulative_roughage_fed +
    oct_roughage_fed
Else if month == November
    nov_roughage_fed = nov_roughage_supply * (1 – feeding_waste_pct)
    cumulative_roughage_fed = cumulative_roughage_fed +
    nov_roughage_fed
Else if month == December
    dec_roughage_fed = dec_roughage_supply * (1 – feeding_waste_pct)
    cumulative_roughage_fed = cumulative_roughage_fed +
    dec_roughage_fed

#Total roughage feed this year
this_year_roughage_fed = cumulative_roughage_fed

    #Reset cumulative roughage feed to zero for next year
    cumulative_roughage_fed = 0.00

#Send monthly roughage feed amounts for this year to Output
For grazing_system_year
    output grazing_system_year, jan_roughage_fed, feb_roughage_fed,
    mar_roughage_fed, apr_roughage_fed, may_roughage_fed,
    jun_roughage_fed, jul_roughage_fed, aug_roughage_fed,
    sep_roughage_fed, oct_roughage_fed, nov_roughage_fed,
    dec_roughage_fed, this_year_roughage_fed


#Calculate monthly forage supply including roughage fed for each year
For each year in the grazing system
    jan_forage_roughage_supply = jan_forage_supply + jan_roughage_fed
    feb_forage_roughage_supply = feb_forage_supply + feb_roughage_fed
    mar_forage_roughage_supply = mar_forage_supply + mar_roughage_fed
    apr_forage_roughage_supply = apr_forage_supply + apr_roughage_fed
    may_forage_roughage_supply = may_forage_supply + may_roughage_fed
    jun_forage_roughage_supply = jun_forage_supply + jun_roughage_fed
    jul_forage_roughage_supply = jul_forage_supply + jul_roughage_fed
    aug_forage_roughage_supply = aug_forage_supply + aug_roughage_fed
    sep_forage_roughage_supply = sep_forage_supply + sep_roughage_fed
```

oct_forage_roughage_supply = oct_forage_supply + oct_roughage_fed
nov_forage_roughage_supply = nov_forage_supply + npv_roughage_fed
dec_forage_roughage_supply = dec_forage_supply + dec_roughage_fed

#Calculate forage supply including roughage fed for this year
this_year_forage_roughage_supply = this_year_forage_supply +
this_year_roughage_fed

#Send monthly forage supply include roughage fed amounts for this year to Output
For grazing_system_year
output grazing_system_year, jan_forage_roughage_supply,
feb_forage_roughage_supply, mar_forage_roughage_supply,
apr_forage_roughage_supply, may_forage_roughage_supply,
jun_forage_roughage_supply, jul_forage_roughage_supply,
aug_forage_roughage_supply, sep_forage_roughage_supply,
oct_forage_roughage_supply, nov_forage_roughage_supply,
dec_forage_roughage_supply, this_year_forage_roughage_supply

#Update system forage supply including roughage fed
system_forage_roughage_supply = system_forage_roughage_supply +
this_year_forage_roughage_supply

#Calculate monthly forage animal balances for this grazing system year
For each grazing_system_year
fab_january = jan_forage_supply + jan_roughage_fed – jan_system_demand
fab_february = feb_forage_supply + feb_roughage_fed – feb_system_demand
fab_march = mar_forage_supply + mar_roughage_fed – mar_systemdemand
fab_april = apr_forage_supply + apr_roughage_fed – apr_system_demand
fab_may = may_forage_supply + may_roughage_fed – may_system_demand
fab_june = jun_forage_supply + jun_roughage_fed – jun_system_demand
fab_july = jul_forage_supply + jul_roughage_fed – jul_system_demand
fab_aug = aug_forage_supply + aug_roughage_fed – aug_system_demand
fab_september = sep_forage_supply + sep_roughage_fed – sep_system_demand
fab_october = oct_forage_supply + oct_roughage_fed – oct_system_demand
fab_november = nov_forage_supply + nov_roughage_fed – nov_system_demand
fab_december = dec_forage_supply + dec_roughage_fed – dec_system_demand

#Calculate forage animal balance for this year
this_year_forage_animal_balance = this_year_forage_supply +
this_year_roughage_fed – total_system_demand

#Update system forage animal balance
system_forage_animal_balance = system_forage_animal_balance +
this_year_forage_animal_balance

#Send monthly forage animal balances and forage animal balance for this year to Output
For this grazing_system_year
Output grazing_system_year, fab_january, fab_february, fab_march, fab_april, fab_may, fab_june, fab_july, fab_august, fab_september, fab_october, fab_november, fab_december, this_year_forage_animal_balance

#Pass AoA monthly and total forage supplies for this year from Input to Output
Output jan_aoa_forage_supply, feb_aoa_forage_supply, mar_aoa_forage_supply, apr_aoa_forage_supply, may_aoa_forage_supply, jun_aoa_forage_supply, jul_aoa_forage_supply, aug_aoa_forage_supply
sep_aoa_forage_supply, oct_aoa_forage_supply, nov_aoa_forage_supply, dec_aoa_forage_supply, this_year_aoa_forage_supply

#Pass AoA acres from Input to Output
Output aoa_acres

#Pass monthly forage supplies for this year from Input to Output
Ouput jan_forage_supply, feb_forage_supply, mar_forage_supply, apr_forage_supply, may_forage_supply, jun_forage_supply, jul_forage_supply, aug_forage_supply, sep_forage_supply, oct_forage_supply, nov_forage_supply, dec_forage_supply, this_year_forage_supply

#Pass herd monthly and total demands for this year from Input to Output
Output herd_id, jan_herd_demand, feb_herd_demand, mar_herd_demand, apr_herd_demand, may_herd_demand, jun_herd_demand, jul_herd_demand, aug_herd_demand, sep_herd_demand, oct_herd_demand, nov_herd_demand, dec_herd_demand, total_herd_demand

#Pass system monthly and total demands for this year from Input to Output
Output jan_system_demand, feb_system_demand, mar_system_demand, apr_system_demand, may_system_demand, jun_system_demand, jul_system_demand, aug_system_demand, sep_system_demand, oct_system_demand, nov_system_demand, dec_system_demand, total_system_demand

#Update system animal demand
system_animal_demand = system_animal_demand + total_system_demand

#Send system forage animal balance to Output
Output system_forage_animal_balance, system_forage_roughage_supply, system_animal_demand

1.3. Output
system_forage_animal_balance

system_forage_roughage_supply
system_animal_demand
grazing_system_year … one or more in the grazing system
AoA identifier … one or more in the grazing system
aoa_acres
jan_aoa_forage_supply
feb_aoa_forage_supply
mar_aoa_forage_supply
apr_aoa_forage_supply
may_aoa_forage_supply
jun_aoa_forage_supply
jul_aoa_forage_supply
aug_aoa_forage_supply
sep_aoa_forage_supply
oct_aoa_forage_supply
nov_aoa_forage_supply
dec_aoa_forage_supply
this_year_aoa_forage_supply

jan_forage_supply
feb_forage_supply
mar_forage_supply
apr_forage_supply
may_forage_supply
jun_forage_supply
jul_forage_supply
aug_forage_supply
sep_forage_supply
oct_forage_supply
nov_forage_supply
dec_forage_supply
this_year_forage_supply

jan_roughage_fed
feb_roughage_fed
mar_roughage_fed
apr_roughage_fed
may_roughage_fed
jun_roughage_fed
jul_roughage_fed
aug_roughage_fed
sep_roughage_fed
oct_roughage_fed
nov_roughage_fed
dec_roughage_fed
this_year_roughage_fed

jan_forage_roughage_supply
feb_forage_roughage_supply
mar_forage_roughage_supply
apr_forage_roughage_supply
may_forage_roughage_supply
jun_forage_roughage_supply
jul_forage_roughage_supply
aug_forage_roughage_supply
sep_forage_roughage_supply
oct_forage_roughage_supply
nov_forage_roughage_supply
dec_forage_roughage_supply
this_year_forage_roughage_supply

by herd_id
    jan_herd_demand
    feb_herd_demand
    mar_herd_demand
    apr_herd_demand
    may_herd_demand
    jun_herd_demand
    jul_herd_demand
    aug_herd_demand
    sep_herd_demand
    oct_herd_demand
    nov_herd_demand
    dec_herd_demand
    this_year_herd_demand

jan_system_demand
feb_system_demand
mar_system_demand
apr_system_demand
may_system_demand
jun_system_demand
jul_system_demand
aug_system_demand
sep_system_demand
oct_system_demand
nov_system_demand
dec_system_demand
total_system_demand

fab_january
fab_february
fab_march

fab_april
fab_may
fab_june
fab_july
fab_august
fab_september
fab_october
fab_november
fab_december
this_year_forage_animal_balance

Service GRAS-13: Calculate Monthly, Yearly, and System Forage Animal Balance With Grazing Schedule (CalcDetailFAB)

Purpose: Calculate monthly, yearly, and system forage animal balances (FABs) for a grazing system including a grazing schedule. Forage supply and animal demand are calculated daily through all years in the grazing system, accounting for harvested roughage, roughage additions, forage partition profile activities and harvest efficiencies. Herds are scheduled onto areas of analysis (AoAs) during always available forage partition profile (FPP) activity periods. The primary difference between this service and GRAS-11 (without grazing schedule) centers on calculating FABs for each AoA in the grazing system rather than at the system level.

Service Signature

Request Payload

grazing_system_start_date … date, e.g. January 1, 2016 (in suitable date format); Grazing System Start Date
grazing_system_end_date… date; e.g. December 31, 2020 (in suitable date format); Grazing System End Date

#Input data for calculating daily animal demand; should be sufficient number of herds and groups for the grazing expected during the entire period of the grazing system, mediated by date in and out of herd.
herd_id … integer; one or more in the grazing system; Herd Identifier
animal_group_id … integer; one or more in the herd; Animal Group Identifier
number_of_head … integer; Number of Head in Group
average_weight_lbs … integer; Average Weight of Animal in Pounds
avg_daily_intake_pct … decimal(4,3); Percent Average Daily Intake of Animal
date_into_herd … date; Date Animal Group enters Herd
date_out_of_herd … date; Date Animal Group leaves Herd

#Input data for herd assignment to AoAs during always available FPP activity periods
AoAId … integer; one or more in the grazing system; Area of Analysis Identifier
aoa_acres … decimal (10,2); Area of Analysis Acres
graze_period_id … integer; one or more on the AoA; Graze Period Identifier
herd_id … integer; one or more in the grazing period, Herd Identifier
graze_start_date … date, must be within always available FPP activity; Date Herd enters Graze Period
graze_end_date … date, must be within always available FPP activity; Date Herd leaves Graze Period

#Annual total and monthly forage supplies for the grazing system
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
this_year_forage_supply … numeric (10,1); This Year Forage Supply
jan_forage_supply … numeric (9,1); January Forage Supply
feb_forage_supply … numeric (9,1); February Forage Supply

mar_forage_supply … numeric (9,1); March Forage Supply
apr_forage_supply … numeric (9,1); April Forage Supply
may_forage_supply … numeric (9,1); May Forage Supply
jun_forage_supply … numeric (9,1); June Forage Supply
jul_forage_supply … numeric (9,1); July Forage Supply
aug_forage_supply … numeric (9,1); August Forage Supply
sep_forage_supply … numeric (9,1); September Forage Supply
oct_forage_supply … numeric (9,1); October Forage Supply
nov_forage_supply … numeric (9,1); November Forage Supply
dec_forage_supply … numeric (9,1); December Forage Supply

#AoA monthly forage supply and roughage fed for each year in the grazing system

AoAId … integer; one or more in the grazing system; Area of Analysis Identifier
grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
jan_aoa_forage_supply … numeric (9,1); January AoA Forage Supply
feb_aoa_forage_supply … numeric (9,1); February AoA Forage Supply
mar_aoa_forage_supply … numeric (9,1); March AoA Forage Supply
apr_aoa_forage_supply … numeric (9,1); April AoA Forage Supply
may_aoa_forage_supply … numeric (9,1); May AoA Forage Supply
jun_aoa_forage_supply … numeric (9,1); June AoA Forage Supply
jul_aoa_forage_supply … numeric (9,1); July AoA Forage Supply
aug_aoa_forage_supply … numeric (9,1); August AoA Forage Supply
sep_aoa_forage_supply … numeric (9,1); September AoA Forage Supply
oct_aoa_forage_supply … numeric (9,1); October AoA Forage Supply
nov_aoa_forage_supply … numeric (9,1); November AoA Forage Supply
dec_aoa_forage_supply … numeric (9,1); December AoA Forage Supply
this_year_aoa_forage_supply … numeric (10,1); This Year AoA Forage Supply

jan_aoa_roughage_supply … numeric (7,0); January Rouhage Supply
feb_aoa_roughage_supply … numeric (7,0); February Rouhage Supply
mar_aoa_roughage_supply … numeric (7,0); March Rouhage Supply
apr_aoa_roughage_supply … numeric (7,0); April Rouhage Supply
may_aoa_roughage_supply … numeric (7,0); May Rouhage Supply
jun_aoa_roughage_supply … numeric (7,0); June Rouhage Supply
jul_aoa_roughage_supply … numeric (7,0); July Rouhage Supply
aug_aoa_roughage_supply … numeric (7,0); August Rouhage Supply
sep_aoa_roughage_supply … numeric (7,0); September Rouhage Supply
oct_aoa_roughage_supply … numeric (7,0); October Rouhage Supply
nov_aoa_roughage_supply … numeric (7,0); November Rouhage Supply
dec_aoa_roughage_supply … numeric (7,0); December Rouhage Supply

#Feeding waste percentage that is utilized to adjust amount of roughage fe-ed each month each year.

grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System
feeding_waste_pct … numeric (3,1); Percent of Feeding Waste

Result Payload

system_forage_animal_balance … numeric (10,1); Grazing System Forage Animal Balance
system_forage_roughage_supply … numeric (10,1); Grazing System Forage Supply including Roughage Fed
system_animal_demand … numeric (10,1); Grazing System Animal Demand
AoAId … integer; one or more in the grazing system; Area of Analysis Identifier
aoa_acres … decimal (10,2); Area of Analysis Acres
grazing_system_year … smallint (e.g. 2016); one or more for each AoA in the grazing system; Year in the Grazing System
jan_aoa_forage_supply … numeric (9,1); January AoA Forage Supply
feb_aoa_forage_supply … numeric (9,1); February AoA Forage Supply
mar_aoa_forage_supply … numeric (9,1); March AoA Forage Supply
apr_aoa_forage_supply … numeric (9,1); April AoA Forage Supply
may_aoa_forage_supply … numeric (9,1); May AoA Forage Supply
jun_aoa_forage_supply … numeric (9,1); June AoA Forage Supply
jul_aoa_forage_supply … numeric (9,1); July AoA Forage Supply
aug_aoa_forage_supply … numeric (9,1); August AoA Forage Supply
sep_aoa_forage_supply … numeric (9,1); September AoA Forage Supply
oct_aoa_forage_supply … numeric (9,1); October AoA Forage Supply
nov_aoa_forage_supply … numeric (9,1); November AoA Forage Supply
dec_aoa_forage_supply … numeric (9,1); December AoA Forage Supply
this_year_aoa_forage_supply … numeric (10,1); This Year AoA Forage Supply

jan_aoa_roughage_fed … numeric (8,1); January AoA Rouhage Fed
feb_aoa_roughage_fed … numeric (8,1); February AoA Rouhage Fed
mar_aoa_roughage_fed … numeric (8,1); March AoA Rouhage Fed
apr_aoa_roughage_fed … numeric (8,1); April AoA Rouhage Fed
may_aoa_roughage_fed … numeric (8,1); May AoA Rouhage Fed
jun_aoa_roughage_fed … numeric (8,1); June AoA Rouhage Fed
jul_aoa_roughage_fed … numeric (8,1); July AoA Rouhage Fed
aug_aoa_roughage_fed … numeric (8,1); August AoA Rouhage Fed
sep_aoa_roughage_fed … numeric (8,1); September AoA Rouhage Fed
oct_aoa_roughage_fed … numeric (8,1); October AoA Rouhage Fed
nov_aoa_roughage_fed … numeric (8,1); November AoA Rouhage Fed
dec_aoa_roughage_fed … numeric (8,1); December AoA Rouhage Fed
this_year_aoa_roughage_fed … numeric (9,1); This Year AoARouhage Fed

jan_aoa_animal_demand … numeric (9,1); January AoA Animal Demand
feb_aoa_animal_demand … numeric (9,1); February AoA Animal Demand
mar_aoa_animal_demand … numeric (9,1); March AoA Animal Demand
apr_ aoa_animal_demand … numeric (9,1); April AoA Animal Demand

may_ aoa_animal_demand … numeric (9,1); May AoA Animal Demand
jun_ aoa_animal_demand … numeric (9,1); June AoA Animal Demand
jul_ aoa_animal_demand … numeric (9,1); July AoA Animal Demand
aug_ aoa_animal_demand … numeric (9,1); August AoA Animal Demand
sep_ aoa_animal_demand … numeric (9,1); September AoA Animal Demand
oct_ aoa_animal_demand … numeric (9,1); October AoA Animal Demand
nov_ aoa_animal_demand … numeric (9,1); November AoA Animal Demand
dec_ aoa_animal_demand … numeric (9,1); December AoA Animal Demand
this_year_ aoa_animal_demand … numeric (10,1); This Year AoA Animal Demand

fab_ aoa_january … numeric (9,1); January AoA Forage Animal Balance
fab_ aoa_february … numeric (9,1); February AoA Forage Animal Balance
fab_ aoa_march … numeric (9,1); March AoA Forage Animal Balance
fab_ aoa_april … numeric (9,1); April AoA Forage Animal Balance
fab_ aoa_may … numeric (9,1); May AoA Forage Animal Balance
fab_ aoa_june … numeric (9,1); June AoA Forage Animal Balance
fab_ aoa_july … numeric (9,1); July AoA Forage Animal Balance
fab_ aoa_august … numeric (9,1); August AoA Forage Animal Balance
fab_ aoa_september … numeric (9,1); September AoA Forage Animal Balance
fab_ aoa_october … numeric (9,1); October AoA Forage Animal Balance
fab_ aoa_november … numeric (9,1); November AoA Forage Animal Balance
fab_ aoa_december … numeric (9,1); December AoA Forage Animal Balance
this_year_ aoa_forage_animal_balance … numeric (10,1); This Year AoA Forage Animal Balance

grazing_system_year … one or more in the grazing system
jan_forage_supply … numeric (9,1); January Forage Supply
feb_forage_supply … numeric (9,1); February Forage Supply
mar_forage_supply … numeric (9,1); March Forage Supply
apr_forage_supply … numeric (9,1); April Forage Supply
may_forage_supply … numeric (9,1); May Forage Supply
jun_forage_supply … numeric (9,1); June Forage Supply
jul_forage_supply … numeric (9,1); July Forage Supply
aug_forage_supply … numeric (9,1); August Forage Supply
sep_forage_supply … numeric (9,1); September Forage Supply
oct_forage_supply … numeric (9,1); October Forage Supply
nov_forage_supply … numeric (9,1); November Forage Supply
dec_forage_supply … numeric (9,1); December Forage Supply
this_year_forage_supply … numeric (10,1); This Year Forage Supply

jan_roughage_fed … numeric (8,1); January Rouhage Fed
feb_roughage_fed … numeric (8,1); February Rouhage Fed
mar_roughage_fed … numeric (8,1); March Rouhage Fed
apr_roughage_fed … numeric (8,1); April Rouhage Fed
may_roughage_fed … numeric (8,1); May Rouhage Fed
jun_roughage_fed … numeric (8,1); June Rouhage Fed

jul_roughage_fed … numeric (8,1); July Rouhage Fed
aug_roughage_fed … numeric (8,1); August Rouhage Fed
sep_roughage_fed … numeric (8,1); September Rouhage Fed
oct_roughage_fed … numeric (8,1); October Rouhage Fed
nov_roughage_fed … numeric (8,1); November Rouhage Fed
dec_roughage_fed … numeric (8,1); December Rouhage Fed
this_year_roughage_fed… numeric (9,1); This Year Rouhage Fed

jan_forage_roughage_supply … numeric (10,1); January Forage Supply Including Rouhage Fed
feb_forage_roughage_supply … numeric (10,1); February Forage Supply Including Rouhage Fed
mar_forage_roughage_supply … numeric (10,1); March Forage Supply Including Rouhage Fed
apr_forage_roughage_supply … numeric (10,1); April Forage Supply Including Rouhage Fed
may_forage_roughage_supply … numeric (10,1); May Forage Supply Including Rouhage Fed
jun_forage_roughage_supply… numeric (10,1); June Forage Supply Including Rouhage Fed
jul_forage_roughage_supply … numeric (10,1); July Forage Supply Including Rouhage Fed
aug_forage_roughage_supply … numeric (10,1); August Forage Supply Including Rouhage Fed
sep_forage_roughage_supply … numeric (10,1); September Forage Supply Including Rouhage Fed
oct_forage_roughage_supply … numeric (10,1); October Forage Supply Including Rouhage Fed
nov_forage_roughage_supply … numeric (10,1); November Forage Supply Including Rouhage Fed
dec_forage_roughage_supply … numeric (10,1); December Forage Supply Including Rouhage Fed
this_year_forage_roughage_supply … numeric (10,1); This Year Forage Supply Including Rouhage Fed

jan_animal_demand … numeric (9,1); January Animal Demand
feb_animal_demand … numeric (9,1); February Animal Demand
mar_animal_demand … numeric (9,1); March Animal Demand
apr_animal_demand … numeric (9,1); April Animal Demand
may_animal_demand … numeric (9,1); May Animal Demand
jun_animal_demand … numeric (9,1); June Animal Demand
jul_animal_demand … numeric (9,1); July Animal Demand
aug_animal_demand … numeric (9,1); August Animal Demand
sep_animal_demand … numeric (9,1); September Animal Demand
oct_animal_demand … numeric (9,1); October Animal Demand
nov_animal_demand … numeric (9,1); November Animal Demand
dec_animal_demand … numeric (9,1); Decemer Animal Demand

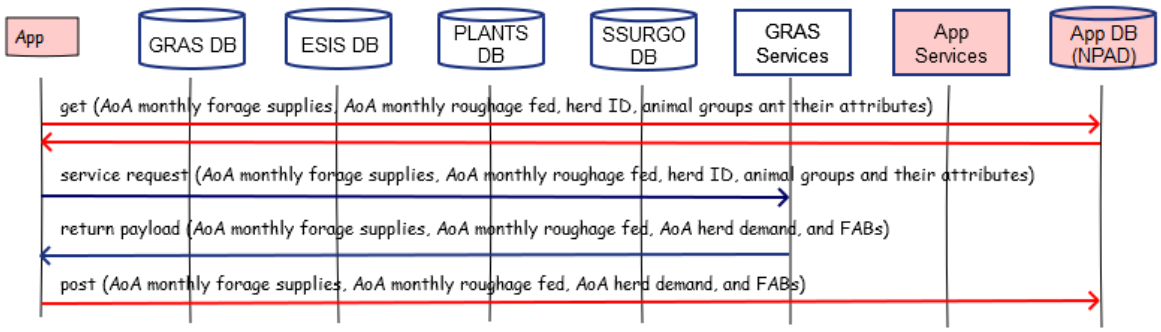
this_year_animal_demand … numeric (10,1); This Year Animal Demand

fab_january … numeric (9,1); January Forage Animal Balance
fab_february … numeric (9,1); February Forage Animal Balance
fab_march … numeric (9,1); March Forage Animal Balance
fab_april … numeric (9,1); April Forage Animal Balance
fab_may … numeric (9,1); May Forage Animal Balance
fab_june … numeric (9,1); June Forage Animal Balance
fab_july … numeric (9,1); July Forage Animal Balance
fab_august … numeric (9,1); August Forage Animal Balance
fab_september … numeric (9,1); September Forage Animal Balance
fab_october … numeric (9,1); October Forage Animal Balance
fab_november … numeric (9,1); November Forage Animal Balance
fab_december … numeric (9,1); December Forage Animal Balance
this_year_forage_animal_balance … numeric (10,1); This Year Forage Animal Balance

Reference Data Sources

None accessed by this service

GRAS-13: Calculate Forage Animal Balance With Grazing Schedule



Component

1. **Calculate Animal Demand for AoAs in Grazing System (CalcAoAAniDemand)**
   1.1. Inputs
   grazing_system_start_date … e.g. January 1, 2016 (in suitable date format)
   grazing_system_end_date… e.g. December 31, 2020 (in suitable date format)
   herd_id … one or more in the grazing system
   animal_group_id … one or more in the herd
   number_of_head
   average_weight_lbs
   avg_daily_intake_pct
   date_into_herd
   date_out_of_herd
   AoAID … one or more in the grazing system
   aoa_acres

graze_period_id … one or more on the AoA
herd_id … one or more in the grazing period
graze_start_date … must be within always available FPP activity
graze_end_date … must be within always available FPP activity

1.2. Methods

#Calculate monthly and yearly animal demand for the AoAs in the grazing system

For each day (date) in the grazing system

#Set this day, month of this day, and year of this day

this_day = date in year of the grazing system
If leap year (e.g. 2016, 2020, 2024, etc.)
    this_month = month of this_day in the leap year
Else
    this_month = month of this_day in the non-leap year
grazing_system_year = year in the grazing system (e.g. 2018)

#Calculate animal demand for each AoA on this day

For each AoA in the grazing system on this day
    this_aoa = AoAID

#Set initial animal demand for this AoA on first day of grazing system

If this_day == grazing_system_start_date
    aoa_animal_demand = 0.00
    prev_year_aoa_animal_demand = 0.00

#Calculate animal demand for this AoA on this day

For each graze_period_id in AoA (from request payload)
    For each herd_id in the graze period
        If this_day >= graze_start_date and <= graze_end_date
            For each animal_group_id in the herd
                this_day_animal_demand = this_day_animal_demand +
                (number_of_head * average_weight_lbs *
                avg_daily_intake_pct)

#Update cumulative animal demand for this AoA on this day in the grazing system

aoa_animal_demand = aoa_animal_demand + this_day_animal_demand

#Update total cumulative animal demand for the system on this day in the grazing system

cumulative_animal_demand = cumulative_animal_demand +
this_day_animal_demand

#Reset this AoA's daily animal demand to zero for the next AoA

this_day_animal_demand = 0.00

#Update this AoA's monthly animal demands on this day in this year

```
If this_day == January 31st
    jan_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    jan_animal_demand = jan_animal_demand + jan_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if leap year and this_day == February 29th
    feb_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    feb_animal_demand = feb_animal_demand + feb_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == February 28th
    feb_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    mar_animal_demand = mar_animal_demand + mar_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == March 31st
    mar_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    apr_animal_demand = apr_animal_demand + apr_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == April 30th
    apr_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    may_animal_demand = may_animal_demand +
    may_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == May 31st
    may_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    may_animal_demand = may_animal_demand +
    may_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == June 30th
    jun_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    jun_animal_demand = jun_animal_demand + jun_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == July 31st
    jul_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    jul_animal_demand = jul_animal_demand + jul_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == August 31st
    aug_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    aug_animal_demand = aug_animal_demand + aug_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
```

Else if this_day == September 30th
    sep_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    sep_animal_demand = sep_animal_demand + sep_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == October 31st
    oct_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    oct_animal_demand = oct_animal_demand + oct_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == November 30th
    nov_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    nov_animal_demand = nov_animal_demand + nov_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand
Else if this_day == December 31st (last day of this year)
    dec_aoa_animal_demand = aoa_animal_demand –
    prev_aoa_animal_demand
    dec_animal_demand = dec_animal_demand + dec_aoa_animal_demand
    prev_aoa_animal_demand = aoa_animal_demand

    #Calculate AoA animal demand for this year
    this_year_aoa_animal_demand = aoa_animal_demand –
    prev_year_aoa_animal_demand
    prev_year_aoa_animal_demand = aoa_animal_demand

    #Send AoA monthly animal demands for this year to Output
    For this_year and this_aoa
        output jan_aoa_animal_demand, feb_aoa_animal_demand,
        mar_aoa_animal_demand, apr_aoa_animal_demand,
        may_aoa_animal_demand, jun_aoa_animal_demand,
        jul_aoa_animal_demand, aug_aoa_animal_demand,
        sep_aoa_animal_demand, oct_aoa_animal_demand,
        nov_aoa_animal_demand, dec_aoa_animal_demand,
        this_year_aoa_animal_demand

#Persist and pass today's monthly AoA animal demand to next day iteration
Pass to next day: this_aoa, jan_aoa_animal_demand,
feb_aoa_animal_demand, mar_aoa_animal_demand,
apr_aoa_animal_demand, may_aoa_animal_demand,
jun_aoa_animal_demand, jul_aoa_animal_demand,
aug_aoa_animal_demand, sep_aoa_animal_demand,
oct_aoa_animal_demand, nov_aoa_animal_demand,
dec_aoa_animal_demand, this_year_aoa_animal_demand

#Persist and pass today's monthly total animal demand to next day iteration
Pass to next day: jan_animal_demand, feb_animal_demand,

mar_animal_demand, apr_animal_demand, may_animal_demand,
jun_animal_demand, jul_animal_demand, aug_animal_demand,
sep_animal_demand, oct_animal_demand, nov_animal_demand,
dec_animal_demand

#Set total animal demand for this year in the grazing system
If this_day == December 31st (last day of this grazing_system year)
this_year_animal_demand = cumulative_animal_demand

#Send total and monthly animal demand for this year to Output
Output grazing_system_year, this_year_animal_demand,
jan_animal_demand, feb_animal_demand, mar_animal_demand,
apr_animal_demand, may_animal_demand, jun_animal_demand,
jul_animal_demand, aug_animal_demand, sep_animal_demand,
oct_animal_demand, nov_animal_demand, dec_animal_demand

#Set total animal demand for the grazing system
If this_day == grazing_system_end_date
system_demand = cumulative_animal_demand

#Send to Output
Output _system_animal_demand

#Pass AoA acres from Input to Output
Output aoa_acres

1.3. Output
system_animal_demand
grazing_system_year … one or more
jan_animal_demand
feb_animal_demand
mar_animal_demand
apr_animal_demand
may_animal_demand
jun_animal_demand
jul_animal_demand
aug_animal_demand
sep_animal_demand
oct_animal_demand
nov_animal_demand
dec_animal_demand
this_year_animal_demand

AoAId … one or more in the grazing system
aoa_acres
jan_aoa_animal_demand
feb_aoa_animal_demand

mar_aoa_animal_demand
apr_aoa_animal_demand
may_aoa_animal_demand
jun_aoa_animal_demand
jul_aoa_animal_demand
aug_aoa_animal_demand
sep_aoa_animal_demand
oct_aoa_animal_demand
nov_aoa_animal_demand
dec_aoa_animal_demand
this_year_aoa_animal_demand

2. <u>Calculate Detailed Forage Animal Balance (CalcDetailedFAB)</u>

2.1. Inputs

AoAId … one or more in the grazing system

grazing_system_year … one or more

#From request payload

jan_aoa_forage_supply
feb_aoa_forage_supply
mar_aoa_forage_supply
apr_aoa_forage_supply
may_aoa_forage_supply
jun_aoa_forage_supply
jul_aoa_forage_supply
aug_aoa_forage_supply
sep_aoa_forage_supply
oct_aoa_forage_supply
nov_aoa_forage_supply
dec_aoa_forage_supply
this_year_aoa_forage_supply

jan_aoa_roughage_supply
feb_aoa_roughage_supply
mar_aoa_roughage_supply
apr_aoa_roughage_supply
may_aoa_roughage_supply
jun_aoa_roughage_supply
jul_aoa_roughage_supply
aug_aoa_roughage_supply
sep_aoa_roughage_supply
oct_aoa_roughage_supply
nov_aoa_roughage_supply
dec_aoa_roughage_supply

#From previous component of this service

jan_aoa_animal_demand

feb_aoa_animal_demand
mar_aoa_animal_demand
apr_aoa_animal_demand
may_aoa_animal_demand
jun_aoa_animal_demand
jul_aoa_animal_demand
aug_aoa_animal_demand
sep_aoa_animal_demand
oct_aoa_animal_demand
nov_aoa_animal_demand
dec_aoa_animal_demand
this_year_aoa_animal_demand

grazing_system_year … one or more

#From request payload

this_year_forage_supply
jan_forage_supply
feb_forage_supply
mar_forage_supply
apr_forage_supply
may_forage_supply
jun_forage_supply
jul_forage_supply
aug_forage_supply
sep_forage_supply
oct_forage_supply
nov_forage_supply
dec_forage_supply

#From previous component of this service

jan_animal_demand
feb_animal_demand
mar_animal_demand
apr_animal_demand
may_animal_demand
jun_animal_demand
jul_animal_demand
aug_animal_demand
sep_animal_demand
oct_animal_demand
nov_animal_demand
dec_animal_demand
this_year_animal_demand

#Feeding waste percentage that is utilized to adjust amount of roughage fed each monthduring year.

grazing_system_year … smallint (e.g. 2016); one or more in the grazing system; Year in the Grazing System

feeding_waste_pct

2.2. Methods

#Calculate AOA monthly roughage feed for each year

For each AoA

For each year in the grazing system

If year >= grazing_system_start_date and <=grazing_system_end_date

For each month in the year

If month == January

jan_aoa_roughage_fed = jan_aoa_roughage_supply * (1 – feeding_waste_pct)

cumulative_roughage_fed = cumulative_roughage_fed + jan_aoa_roughage_fed

Else if month == February

feb_aoa_roughage_fed = feb_aoa_roughage_supply * (1 – feeding_waste_pct)

cumulative_roughage_fed = cumulative_roughage_fed + feb_aoa_roughage_fed

Else if month == March

mar_aoa_roughage_fed = mar_aoa_roughage_supply * (1 – feeding_waste_pct)

cumulative_roughage_fed = cumulative_roughage_fed + mar_aoa_roughage_fed

Else if month == April

apr_aoa_roughage_fed = apr_aoa_roughage_supply * (1 – feeding_waste_pct)

cumulative_roughage_fed = cumulative_roughage_fed + apr_aoa_roughage_fed

Else if month == May

may_aoa_roughage_fed = may_aoa_roughage_supply * (1 – feeding_waste_pct)

cumulative_roughage_fed = cumulative_roughage_fed + may_aoa_roughage_fed

Else if month == June

jun_aoa_roughage_fed = jun_aoa_roughage_supply * (1 – feeding_waste_pct)

cumulative_roughage_fed = cumulative_roughage_fed + jun_aoa_roughage_fed

Else if month == July

jul_aoa_roughage_fed = jul_aoa_roughage_supply * (1 – feeding_waste_pct)

cumulative_roughage_fed = cumulative_roughage_fed + jul_aoa_roughage_fed

Else if month == August

aug_aoa_roughage_fed = aug_aoa_roughage_supply * (1 –

feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed + aug_aoa_roughage_fed
Else if month == September
sep_aoa_roughage_fed = sep_aoa_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed + sep_aoa_roughage_fed
Else if month == October
oct_aoa_roughage_fed = oct_aoa_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed + oct_aoa_roughage_fed
Else if month == November
nov_aoa_roughage_fed = nov_aoa_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed + nov_aoa_roughage_fed
Else if month == December
dec_aoa_roughage_fed = dec_aoa_roughage_supply * (1 – feeding_waste_pct)
cumulative_roughage_fed = cumulative_roughage_fed + dec_aoa_roughage_fed

#Total roughage feed for AoA this year
this_year_aoa_roughage_fed = cumulative_roughage_fed

#Reset cumulative roughage feed to zero for next year
cumulative_roughage_fed = 0.00

#Send monthly AoA roughage feed amounts for this year to Output
output grazing_system_year, AoAId, jan_aoa_roughage_fed, feb_aoa_roughage_fed, mar_aoa_roughage_fed, apr_aoa_roughage_fed, may_aoa_roughage_fed, jun_aoa_roughage_fed, jul_aoa_roughage_fed, aug_aoa_roughage_fed, sep_aoa_roughage_fed, oct_aoa_roughage_fed, nov_aoa_roughage_fed, dec_aoa-roughage_fed, this_year_aoa_roughage_fed

For each grazing_system_year
#Calculate AOA monthly forage animal balance for this year
fab_aoa_january = jan_aoa_forage_supply + jan_aoa_roughage_fed – jan_aoa_animal_demand
fab_aoa_february = feb_aoa_forage_supply + feb_aoa_roughage_fed – feb_ aoa_animal_demand
fab_aoa_march = mar_aoa_forage_supply + mar_aoa_roughage_fed – mar_aoa_animal_demand
fab_aoa_april = apr_aoa_forage_supply + apr_aoa_roughage_fed –

apr_aoa_animal_demand
fab_aoa_may = may_aoa_forage_supply + may_aoa_roughage_fed – may_aoa_animal_demand
fab_aoa_june = jun_aoa_forage_supply + jun_aoa_roughage_fed – jun_aoa_animal_demand
fab_aoa_july = jul_aoa_forage_supply + jul_aoa_roughage_fed – jul_aoa_animal_demand
fab_aoa_aug = aug_aoa_forage_supply+ aug_aoa_roughage_fed – aug_aoa_animal_demand
fab_aoa_september = sep_aoa_forage_supply + sep_aoa_roughage_fed – sep_aoa_animal_demand
fab_aoa_october = oct_aoa_forage_supply + oct_aoa_roughage_fed – oct_aoa_animal_demand
fab_aoa_november = nov_aoa_forage_supply + nov_aoa_roughage_fed – nov_aoa_animal_demand
fab_aoa_december = dec_aoa_forage_supply + dec_aoa_roughage_fed – dec_aoa_animal_demand

#Calculate AoA annual forage animal balance for this year
this_year_aoa_forage_animal_balance = this_year_aoa_forage_supply + this_year_aoa_roughage_fed – this_year_aoa_animal_demand

#Send AoA monthly forage animal balances and AoA annual forage animal balance for this year to Output
Output grazing_system_year, AoAId, fab_aoa_january, fab_aoa_february, fab_aoa_march, fab_aoa_april, fab_aoa_may, fab_aoa_june, fab_aoa_july, fab_aoa_august, fab_aoa_september, fab_aoa_october, fab_aoa_november, fab_aoa_december, this_year_aoa_forage_animal_balance

#Calculate monthly roughage feed for all AoAs for each year
For each AoA
For each year in the grazing system
For each month in the year
If month == January
jan_roughage_fed = jan_roughage_fed + jan_aoa_roughage_fed
cumulative_roughage_fed = cumulative_roughage_fed + jan_roughage_fed
Else if month == February
feb_roughage_fed = feb_roughage_fed + feb_aoa_roughage_fed
cumulative_roughage_fed = cumulative_roughage_fed + feb_roughage_fed
Else if month == March
mar_roughage_fed = mar_roughage_fed + mar_aoa_roughage_fed
cumulative_roughage_fed = cumulative_roughage_fed + mar_roughage_fed
Else if month == April
apr_roughage_fed = apr_roughage_fed + apr_aoa_roughage_fed

```
    cumulative_roughage_fed = cumulative_roughage_fed +
    apr_roughage_fed
Else if month == May
    may_roughage_fed = may_roughage_fed + may_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed +
    may_roughage_fed
Else if month == June
    jun_roughage_fed = jun_roughage_fed + jun_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed +
    jun_roughage_fed
Else if month == July
    jul_roughage_fed = jul_roughage_fed + jul_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed + jul_roughage_fed
Else if month == August
    aug_roughage_fed = aug_roughage_fed + aug_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed +
    aug_roughage_fed
Else if month == September
    sep_roughage_fed = sep_roughage_fed + sep_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed +
    sep_roughage_fed
Else if month == October
    oct_roughage_fed = oct_roughage_fed + oct_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed +
    oct_roughage_fed
Else if month == November
    nov_roughage_fed = nov_roughage_fed + nov_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed +
    nov_roughage_fed
Else if month == December
    dec_roughage_fed = dec_roughage_fed + dec_aoa_roughage_fed
    cumulative_roughage_fed = cumulative_roughage_fed +
    dec_roughage_fed

    #Total roughage feed for all AoAs this year
        this_year_roughage_fed = cumulative_roughage_fed

        #Reset cumulative roughage feed to zero for next year
        cumulative_roughage_fed = 0.00

#Send monthly roughage feed amounts for all AoAs for this year to Output
output grazing_system_year,  jan_roughage_fed, feb_roughage_fed,
mar_roughage_fed, apr_roughage_fed, may_roughage_fed,
jun_roughage_fed, jul_roughage_fed, aug_roughage_fed, sep_roughage_fed,
oct_roughage_fed, nov_roughage_fed, dec_roughage_fed,
this_year_roughage_fed
```

#Calculate monthly forage supply including roughage fed for all AoAs for each year

For each year in the grazing system

jan_forage_roughage_supply = jan_forage_supply + jan_roughage_fed
feb_forage_roughage_supply = feb_forage_supply + feb_roughage_fed
mar_forage_roughage_supply = mar_forage_supply + mar_roughage_fed
apr_forage_roughage_supply = apr_forage_supply + apr_roughage_fed
may_forage_roughage_supply = may_forage_supply + may_roughage_fed
jun_forage_roughage_supply = jun_forage_supply + jun_roughage_fed
jul_forage_roughage_supply = jul_forage_supply + jul_roughage_fed
aug_forage_roughage_supply = aug_forage_supply + aug_roughage_fed
sep_forage_roughage_supply = sep_forage_supply + sep_roughage_fed
oct_forage_roughage_supply = oct_forage_supply + oct_roughage_fed
nov_forage_roughage_supply = nov_forage_supply + npv_roughage_fed
dec_forage_roughage_supply = dec_forage_supply + dec_roughage_fed

#Calculate forage supply including roughage fed for all AoAs for this year

this_year_forage_roughage_supply = this_year_forage_supply + this_year_roughage_fed

#Send monthly forage supply include roughage fed amounts for this year to Output

For grazing_system_year

output grazing_system_year, jan_forage_roughage_supply, feb_forage_roughage_supply, mar_forage_roughage_supply, apr_forage_roughage_supply, may_forage_roughage_supply, jun_forage_roughage_supply, jul_forage_roughage_supply, aug_forage_roughage_supply, sep_forage_roughage_supply, oct_forage_roughage_supply, nov_forage_roughage_supply, dec_forage_roughage_supply, this_year_forage_roughage_supply

#Update system forage supply

system_forage_roughage_supply = system_forage_roughage_supply + this_year_forage_roughage_supply

#Calculate monthly forage animal balances for this grazing system year

fab_january = jan_forage_supply + jan_roughage_fed – jan_animal_demand
fab_february = feb_forage_supply + feb_roughage_fed - feb_animal_demand
fab_march = mar_forage_supply +mar_roughage_fed – mar_animal_demand
fab_april = apr_forage_supply + apr_roughage_fed – apr_animal_demand
fab_may = may_forage_supply + may_roughage_fed – may_animal_demand
fab_june = jun_forage_supply + jun_roughage_fed – jun_animal_demand
fab_july = jul_forage_supply +jul_roughage_fed – jul_animal_demand
fab_aug = aug_forage_supply +aug_roughage_fed – aug_animal_demand
fab_september = sep_forage_supply +sep_roughage_fed – sep_animal_demand
fab_october = oct_forage_supply + oct_roughage_fed – oct_animal_demand
fab_november = nov_forage_supply + nov_roughage_fed – nov_animal_demand

fab_december = dec_forage_supply + dec_roughage_fed – dec_animal_demand

#Calculate forage animal balance for this year
this_year_forage_animal_balance = this_year_forage_supply + this_year_roughage_fed – this_year_animal_demand
#Send monthly forage animal balances and annual forage animal balance for this year to output
Output grazing system year, fab_january, fab_february, fab_march, fab_april, fab_may, fab_june, fab_july, fab_august, fab_september, fab_october, fab_november, fab_december, this_year_forage_animal_balance

#Update system-wide forage animal balance
system_forage_animal_balance = system_forage_animal_balance + this_year_forage_animal_balance

#Send system forage animal balance to output (after last day in grazing system)
Output system_forage_animal_balance, system_forage_roughage_supply, system_animal_demand

#Pass all forage supply and animal demand Inputs to this service to Outputs
Output
AoA identifier … one or more
grazing_system_year, jan_aoa_forage_supply, feb_aoa_forage_supply, mar_aoa_forage_supply, apr_aoa_forage_supply, may_aoa_forage_supply, jun_aoa_forage_supply, jul_aoa_forage_supply, aug_aoa_forage_supply, sep_aoa_forage_supply, oct_aoa_forage_supply, nov_aoa_forage_supply, dec_aoa_forage_supply, this_year_aoa_forage_supply

grazing_system_year, jan_aoa_animal_demand, feb_aoa_animal_demand, mar_aoa_animal_demand, apr_aoa_animal_demand, may_aoa_animal_demand, jun_aoa_animal_demand, jul_aoa_animal_demand, aug_aoa_animal_demand, sep_aoa_animal_demand, oct_aoa_animal_demand, nov_aoa_animal_demand, dec_aoa_animal_demand, this_year_aoa_animal_demand

grazing_system_year, jan_forage_supply, feb_forage_supply, mar_forage_supply, apr_forage_supply, may_forage_supply, jun_forage_supply, jul_forage_supply, aug_forage_supply, sep_forage_supply, oct_forage_supply, nov_forage_supply, dec_forage_supply, this_year_forage_supply

grazing_system_year, jan_animal_demand, feb_animal_demand, mar_animal_demand, apr_animal_demand, may_animal_demand, jun_animal_demand, jul_animal_demand, aug_animal_demand, sep_animal_demand, oct_animal_demand, nov_animal_demand, dec_animal_demand, this_year_animal_demand

2.3. Output

system_forage_animal_balance
system_forage_roughage_supply
system_animal_demand
AoA identifier … one or more in the grazing system
aoa_acres
grazing_system_year … one or more for each AoA in the grazing system
jan_aoa_forage_supply
feb_aoa_forage_supply
mar_aoa_forage_supply
apr_aoa_forage_supply
may_aoa_forage_supply
jun_aoa_forage_supply
jul_aoa_forage_supply
aug_aoa_forage_supply
sep_aoa_forage_supply
oct_aoa_forage_supply
nov_aoa_forage_supply
dec_aoa_forage_supply
this_year_aoa_forage_supply

jan_aoa_roughage_fed
feb_aoa_roughage_fed
mar_aoa_roughage_fed
apr_aoa_roughage_fed
may_aoa_roughage_fed
jun_aoa_roughage_fed
jul_aoa_roughage_fed
aug_aoa_roughage_fed
sep_aoa_roughage_fed
oct_aoa_roughage_fed
nov_aoa_roughage_fed
dec_aoa_roughage_fed
this_year_aoa_roughage_fed

jan_aoa_animal_demand
feb_aoa_animal_demand
mar_aoa_animal_demand
apr_ aoa_animal_demand
may_ aoa_animal_demand
jun_ aoa_animal_demand
jul_ aoa_animal_demand
aug_ aoa_animal_demand
sep_ aoa_animal_demand
oct_ aoa_animal_demand
nov_ aoa_animal_demand
dec_ aoa_animal_demand

this_year_ aoa_animal_demand

fab_ aoa_january
fab_ aoa_february
fab_ aoa_march
fab_ aoa_april
fab_ aoa_may
fab_ aoa_june
fab_ aoa_july
fab_ aoa_august
fab_ aoa_september
fab_ aoa_october
fab_ aoa_november
fab_ aoa_december
this_year_ aoa_forage_animal_balance

grazing_system_year … one or more for each AoA in the grazing system
jan_forage_supply
feb_forage_supply
mar_forage_supply
apr_forage_supply
may_forage_supply
jun_forage_supply
jul_forage_supply
aug_forage_supply
sep_forage_supply
oct_forage_supply
nov_forage_supply
dec_forage_supply
this_year_forage_supply

jan_roughage_fed
feb_roughage_fed
mar_roughage_fed
apr_roughage_fed
may_roughage_fed
jun_roughage_fed
jul_roughage_fed
aug_roughage_fed
sep_roughage_fed
oct_roughage_fed
nov_roughage_fed
dec_roughage_fed
this_year_roughage_fed

jan_forage_roughage_supply
feb_forage_roughage_supply

mar_forage_roughage_supply
apr_forage_roughage_supply
may_forage_roughage_supply
jun_forage_roughage_supply
jul_forage_roughage_supply
aug_forage_roughage_supply
sep_forage_roughage_supply
oct_forage_roughage_supply
nov_forage_roughage_supply
dec_forage_roughage_supply
this_year_forage_roughage_supply

jan_animal_demand
feb_animal_demand
mar_animal_demand
apr_animal_demand
may_animal_demand
jun_animal_demand
jul_animal_demand
aug_animal_demand
sep_animal_demand
oct_animal_demand
nov_animal_demand
dec_animal_demand
this_year_animal_demand

fab_january
fab_february
fab_march
fab_april
fab_may
fab_june
fab_july
fab_august
fab_september
fab_october
fab_november
fab_december
this_year_forage_animal_balance

Service GRAS-14: Calculate Quick Stocking Rate (CalcQSRate)

Purpose: Calculate a quick estimate of carrying capacity on the grazing units (AoAs) of a grazing system based on unadjusted annual forage production.

Service Signature

Request Payload

AoA identifier … one or more, number of AoAs in the grazing system
- aoa_area … in acres
- aoa_land_use … … integer, corresponding to NRCS land_use_id; choices for this service are 1 – crop, 2 – forest, 3 – range, 4 – pasture, 5 – Protected, 9 – Other Rural Land, and 10 – Associated Agricultural Land
- forage_measured_production … pounds per acre for the AoA

Result Payload

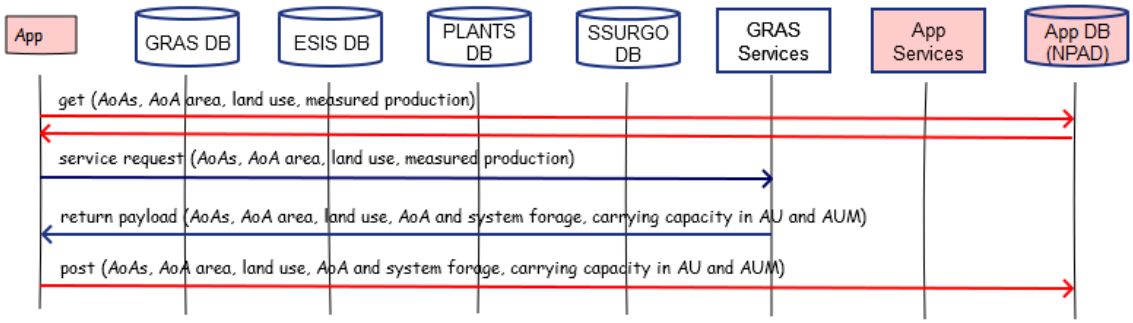
AoA identifier … one or more, number of AoAs in the grazing system
- aoa_area … in acres
- aoa_land_use … integer (1, 2, 3, 4, 5, 9, or 10)
- harvest_efficiency … percent
- grazable_forage … pounds per acre
- available_forage … pounds per acre
- aoa_available_forage … total pounds in the AoA
- carry_capacity_au
- carry_capacity_aum

grazing_system_area .. in acres
system_grazable_forage … weighted average pounds per acre
system_available forage --- weighted average pounds per acre
system_total_forage … sum AoA forage
system_carry_capacity_au
system_carry_capacity_aum

Reference Data Sources

None accessed for this service

GRAS-14: Calculate Quick Stocking Rate



Component

1. Calculate Quick Stocking Rate (CalcQSR)

1.1. Inputs

From request payload

1.2. Methods

#Calculate grazing system area

For each AoA

grazing_system_area = grazing_system_area + aoa_area

For each AoA

If aoa_land_use == 2, 3, 5, 9, or 10

harvest_efficiency = 0.25

Else if 4

harvest_efficiency = 0.30

Else if 1

harvest_efficiency = 0.45

#Output harvest efficiency for this AoA

Output harvest_efficiency for AoA to JSON

#Output AoA acres and land use from Input to Output

Output aoa_area, aoa_land_use for AoA to JSON

#Calculate QSR for this AoA

grazable_forage = forage_measured_production

available_forage = grazable_forage * harvest_efficiency

aoa_available_forage = available_forage * aoa_area

carry_capacity_au = aoa_available_forage / 10950

carry_capacity_aum = carry_capacity_au * 12

#Output QSR for this AoA

Output grazable_forage, available_forage, aoa_available_forage,

carry_capacity_au, carry_capacity_aum for AoA to JSON

#Increment cumulative QSR for grazing system
system_grazable_forage = system_grazable_forage + (grazable_forage * aoa_area / grazing_grazing_system_area)
system_available forage = system_available_forage + (available_forage * aoa_area / grazing_grazing_system_area)
system_total_forage = system_total_forage + aoa_available_forage

#Carry forward cumulative QSR to next AoA
Pass forward system_grazable_forage, system_available forage, system_total_forage to next AoA

#Calculate system carry capacities
system_carry_capacity_au = system_total_forage / 10950
system_carry_capacity_aum = system_carry_capacity_au * 12

#Output system QSR
Output grazable_forage, system_available forage, system_total_forage, system_carry_capacity_au, system_carry_capacity_aum for grazing system to JSON

1.3. Output

AoA identifier … one or more, number of AoAs in the grazing system
- aoa_area … in acres
- aoa_land_use … integer (1, 2, 3, 4, 5, 9, or 10)
- harvest_efficiency … percent
- grazable_forage … pounds per acre
- available_forage … pounds per acre
- aoa_available_forage … total pounds in the AoA
- carry_capacity_au
- carry_capacity_aum

grazing_system_area .. in acres
system_grazable_forage … weighted average pounds per acre
system_available forage --- weighted average pounds per acre
system_total_forage … sum AoA forage
system_carry_capacity_au
system_carry_capacity_aum

Service GRAS-15a: Get Pasture Condition Score Indicators (GetPCSIndicators)

Purpose: Get and return a payload of the pasture condition score indicators to enable the requesting application to complete a pasture condition assessment.

Service Signature

Request Payload

No data is passed into the service for processing other than requesting the service to run

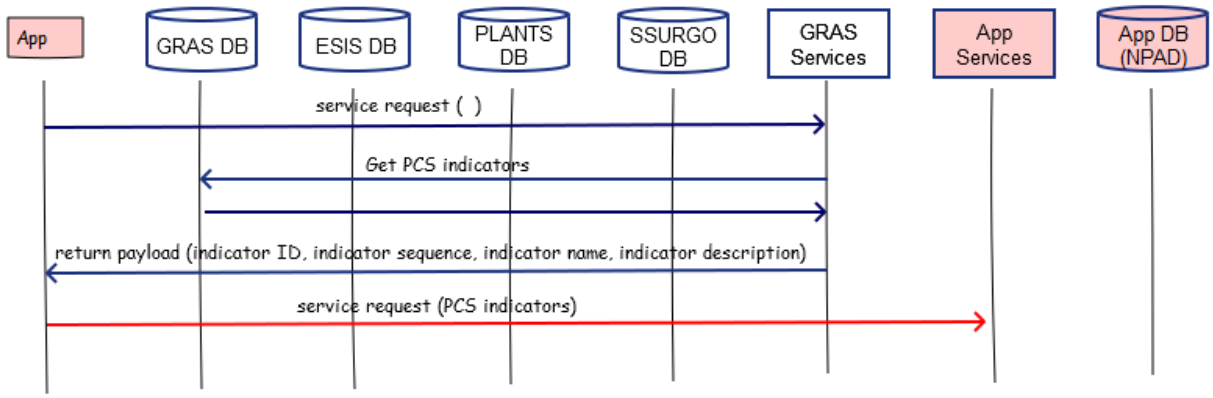
Result Payload

pci_indicators_id … smallint (1, 2, 3, … 14), PCS Indicator Identifier
pci_indicators_sequence … smallint (1, 2, 3, … 14); Display Sequence Number for Indicator
pci_indicators_name … character varying (50); Indicator Name
pci_indicators_description … character varying (254); Indicator Description

Reference Data Sources

GRAS Database
d_pci_indicators

GRAS-15a: Get PCS Indicators



Component

2. Get Pasture Condition Score Indicators (GetPCSIndicators)

2.1. Inputs

2.2. Method

Select

pci_indicators_id
pci_indicators_sequence

pci_indicators_name
pci_indicators_description
From GRAS d_pci_indicators where pci_indicators_obsolete = False

#Send to Output
Output pci_indicators_id, pci_indicators_sequence, pci_indicators_name, pci_indicators_description

2.3. Outputs

pci_indicators_id … smallint (1, 2, 3, … 14), PCS Indicator Identifier
pci_indicators_sequence … smallint (1, 2, 3, … 14); Display Sequence Number for Indicator
pci_indicators_name … character varying (50); Indicator Name
pci_indicators_description … character varying (254); Indicator Description

Service GRAS-15b: Calculate Pasture Condition Score (CalcPCS)

Purpose: Calculate the pasture condition score for a vegetation plot at a particular point in time. Scoring involves the visual evaluation of 10 indicators. Each indicator is rated from 1 (lowest) to 5 (highest) based on descriptive factors for each of the five conditions. The ratings for the 10 indicators are summed into an overall pasture condition score.

Pasture Condition Score Sheet

Farm or ranch site:_____ Date_____

	Pasture Unit Description									
Indicators										
Percent desirable plants[1] Percent plant cover by weight that is desirable forage: 1 <20 · 2 20-40 · 3 40-60 · 4 60-80 · 5 >80										
Plant cover[1] [2] Percent live, leafy canopy cover of desirables and intermediates is: 1 <50 · 2 50-70 · 3 70-90 · 4 90-95 · 5 95-100 Percent live basal area cover of desirables and intermediates is: <15 · 15-25 · 25-35 · 35-50 · >50										
Plant diversity[1] The diversity of well-represented forage species is: 1 · 2 · 3 · 4 · 5 (Read criteria and select appropriate number)										
Plant residue[1] Ground cover, standing dead forage, or thatch is: 1 · 2 · 3 · 4 · 5 (Read criteria and select appropriate number)										
Plant vigor (Read criteria and select appropriate number) Degree of stress of plant community is: 1 · 2 · 3 · 4 · 5 (If less than 4, see Causative factors table. Rate those factors)										
Percent legume[1] [3] Percentage of legume present as total air dry weight: 1 <10, or >60 bloating legume · 2 10-19, or 40-60 spreading legume · 3 20-29 · 4 30-39 · 5 40-60 no grass loss										
Uniformity of use Degree of spot grazing is: 1 >50% ungrazed · 2 25-50% ungrazed · 3 10-25% ungrazed · 4 Minor species rejection · 5 Urine and dung spots ungrazed										
Livestock concentration areas Presence of livestock conc. areas and proximity to surface water: 1 · 2 · 3 · 4 · 5 (Read criteria and select appropriate number)										
Soil compaction Degree of soil compaction is: 1 · 2 · 3 · 4 · 5 (Read criteria and select appropriate number)										
Erosion (Always rate sheet and rill; others only if present) Sheet and rill, and gully, streambank, shoreline, or wind erosion is: 1 Very severe · 2 Severe · 3 Moderate · 4 Slight · 5 No visible										
Pasture condition score										

[1] Pastureland inventory worksheet helpful.
[2] Choose one proper, practical cover type estimation procedure to rate plant cover. The two procedures are not directly comparable.
[3] For warm season grass (C4)-legume stands, use the following criteria: 5, 30-40%; 4, 20-29%; 3, 10-19%; 2, 5-9%, and 1 <4%.

Service Signature

Request Payload

vegetation_plot_id … integer, one or more in the request; Vegetation Plot Identifier
observation_date … date; mm-dd-yyyy; Observation Date for Pasture Condition Score
pci_desirable_plants (rating 1 to 5) … integer; Score for Desirable Plant Species
pci_plant_cover (rating 1 to 5) … integer; Score for Plant Cover
pci_plant_diversity (rating 1 to 5) … integer; Score for Plant Diversity
pci_ground_cover_residue (rating 1 to 5) … integer; Score for Ground Cover Residue
pci_standing_dead_residue (rating 1 to 5) … integer; Score for Standing Dead Residue
pci_plant_vigor (rating 1 to 5) … integer; Score for Plant Vigor
pci_legume_pct_class (rating 1 to 5) … integer; Score for Legume Composition
pci_use_uniformity (rating 1 to 5) … integer; Score for Grazing Uniformity
pci_livestock_conc_areas (rating 1 to 5) … integer; Score for Livestock Concentration Areas
pci_soil_compaction (rating 1 to 5) … integer; Score for Soil Compaction
pci_sheet_rill_erosion (rating 1 to 5) … integer; Score for Sheet/Rill Erosion
pci_wind_erosion (rating 1 to 5) … integer; Score for Wind Erosion
pci_stream_shore_erosion (rating 1 to 5) … integer; Score for Stream/Shoreline Erosion
pci_gully_erosion (rating 1 to 5) … integer; Score for Gully Erosion

Result Payload

vegetation_plot_id … integer; one or more; Vegetation Plot Identifier
observation_date … date, mm-dd-yyyy; Observation Date for Pasture Condition Score
pci_plant_residue_comp_score … numeric(4,2); Plant Residue Composite Score
pci_erosion_comp_score … numeric(4,2); Soil Erosion Composite Score
pci_pasture_condition_score … numeric (4,2); Pasture Condition Score

Reference Data Sources

None accessed for this service

Component

1. **Calculate Pasture Condition Score (CalcPCS)**
   1.1. Inputs

vegetation_plot_id	1	2	3	4	5
pci_desirable_plants	4	3	3	2	5
pci_plant_cover	4	4	4	3	5
pci_plant_diversity	5	3	4	2	3
pci_ground_cover_residue	4	4	4	2	4
pci_standing_dead_residue	2	3	4	2	5
pci_plant_vigor	4	4	4	3	5
pci_legume_pct_class	3	3	4	1	2
pci_use_uniformity	3	3	4	2	5
pci_livestock_conc_areas	4	3	3	2	4
pci_soil_compaction	3	4	4	2	5
pci_sheet_rill_erosion	5	4	5	3	5
pci_wind_erosion				4	
pci_stream_shore_erosion		3			
pci_gully_erosion				4	

   1.2. Methods

#Calculate average plant residue indicator rating

For each vegetation_plot_id
    pci_plant_residue_comp_score = (pci_ground_cover_residue + pci_standing_dead_forage)/2

#Calculate average erosion indicator rating

For each vegetation_plot_id
    If ISNULL(pci_wind_erosion AND pci_stream_shore_erosion AND pci_gully_erosion)
        pci_erosion_comp_score = pci_sheet_rill_erosion
    Else if ISNULL(pci_stream_shore_erosion AND pci_gully_erosion)
        pci_erosion_comp_score = (pci_sheet_rill_erosion + pci_wind_erosion)/2
    Else if ISNULL(pci_wind_erosion AND pci_gully_erosion)
        pci_erosion_comp_score = (pci_sheet_rill_erosion + pci_stream_shore_erosion)/2
    Else if ISNULL(pci_wind_erosion AND pci_stream_shore_erosion)
        pci_erosion_comp_score = (pci_sheet_rill_erosion + pci_gully_erosion)/2
    Else if ISNULL(pci_wind_erosion)
        pci_erosion_comp_score = (pci_sheet_rill_erosion + pci_stream_shore_erosion +pci_gully_erosion)/3
    Else if ISNULL(pci_stream_shore_erosion)
        pci_erosion_comp_score = (pci_sheet_rill_erosion + pci_wind_erosion +pci_gully_erosion)/3
    Else if ISNULL(pci_gully_erosion)
        pci_erosion_comp_score = (pci_sheet_rill_erosion + pci_wind_erosion +pci_stream_shore_erosion)/3

Else

pci_erosion_comp_score = (pci_sheet_rill_erosion + pci_wind_erosion +pci_stream_shore_erosion + pci_gully_erosion)/4

#Calculate PCS for each vegetation plot ID

For each vegetation_plot_id

pci_pasture_condition_score = pci_desirable_plants + pci_plant_cover + pci_plant_diversity + pci_plant_residue_comp_score + pci_plant_vigor + pci_legume_pct_class + pci_use_uniformity + pci_livestock_conc_areas + pci_soil_compaction + pci_erosion_comp_score

1.3. Output

vegetation_plot_id … one or more

observation_date

pci_plant_residue_comp_score

pci_erosion_comp_score

pci_pasture_condition_score

Service GRAS-15c: Get PCS Indicator Rating Choice List (GetPCSChoiceList)

Purpose: Get and return a payload of choice list associated with a specific pasture condition scoring indicator. User selects appropriate rating from the choice list and associated assigned numeric value score for the rating is saved to NPAD.

Example of first three indicators:

Pasture Condition Score Sheet

Indicator	Score 1	2	3	4	5
Percent desirable plants	Desirable species < 20% of stand. Annual weeds and/ or woody species dominant.	Desirable species 20–40% of stand. Mostly weedy annuals and/or woody species present and expanding. Shade a factor.	40–60% desirable forage species. Undesirable broadleaf weeds and annual weedy grasses invading. Some woodies.	60–80% of plant community are desirable species. Remainder mostly intermediates and a few undesirables present.	Desirable species exceed 80% of plant community. Scattered intermediates.
Plant cover (Live stems and green leaf cover of all desirable and intermediate species.)	Canopy: < 50% Basal area: < 15% Photosynthetic area very low. Very little plant cover to slow or stop runoff.	Canopy: 50–70% Basal area: 15–25% Photosynthetic area low. Vegetal retardance to runoff low.	Canopy: 70–90% Basal area: 25–35% Most forages grazed close, little leaf area to intercept sunlight. Moderate vegetal retardance.	Canopy: 90–95% Basal area: 35–50% Spot grazed low and high so some loss of photosynthetic potential. Vegetal retardance still high.	Canopy: 95–100% Basal area: >50% Forages maintained in leafy condition for best photosynthetic activity. Very thick stand, slow or no runoff flows.
Plant diversity	One dominant (> 75% of DM wt.) forage species. Or, over 5 forage species (all <20%) from one dominant functional group, not evenly grazed - poorly distributed.	Two to five forage species from one dominant functional (>75% of DM wt.) group. At least one avoided by livestock permitting presence of mature seed stalks. Species in patches.	Three forage species (each ≥ 20% of DM wt.) from one functional group. None avoided. Or, one forage species each from two functional groups, both supply 25–50% of DM wt.	Three to four forage species (each ≥ 20% of DM wt.) with at least one being a legume. Well intermixed, compatible growth habit, and comparable palatability.	Four to five forage species representing three functional groups (each ≥ 20% of DM wt.) with at least one being a legume. Intermixed well, compatible growth habit, and comparable palatability.

Service Signature

Request Payload

pci_indicators_id … smallint (1, 2, 3, … 14); PCS Indicator Identifier, one per request

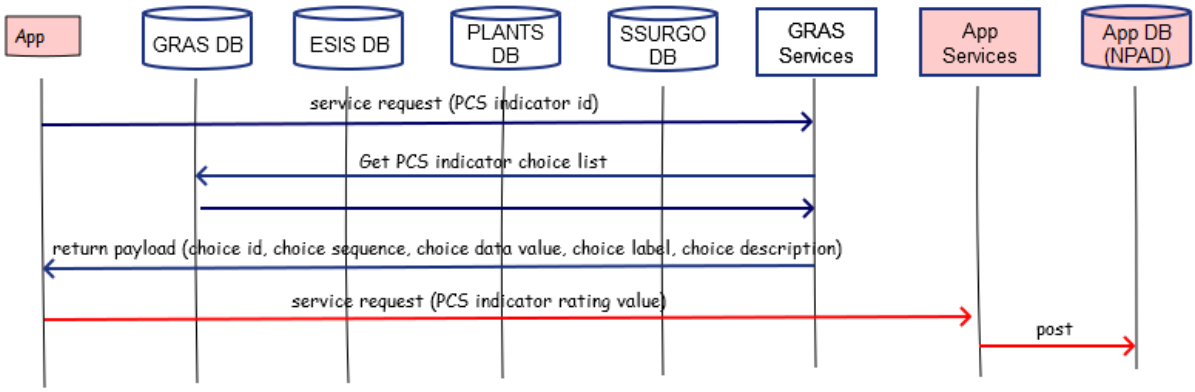
Result Payload

pci_indicators_id … smallint (1, 2, 3, … 14); PCS Indicator Identifier , one per request
choice_id … smallint (1, 2, 3, 4, and 5); Choice Identifier, five per request;
choice_sequence … smallint (1, 2, 3, 4, and 5); Choice Sequence, one per choice_id
choice_data_value … smallint (1, 2, 3, 4, and 5); Choice Data Value, one_per choice_id
choice_label … smallint (1, 2, 3, 4, and 5); Choice Label, one per choice_id
choice_description … character varying (255); Choice Description, one per choice_id

Reference Data Sources

d_pci_indicator_ratings (from GRAS data mart)

GRAS-15c: Get PCS Indicator Rating Choice List



Component

1. **Get PCS Indicator Rating Choice List (GetPCSChoiceList)**
   1.1. Inputs
   pci_indicators_id

   1.2. Methods
   If pci_indicators_id = 1
   Select
   choice_id
   choice_sequence
   choice_data_value
   choice_label
   choice_description
   From d_pci_indicator_ratings where choice_kind_id = 1 AND choice_oboslete = False
   Else if pci_indicators_id = 2
   Select
   choice_id
   choice_sequence
   choice_data_value
   choice_label
   choice_description
   From d_pci_indicator_ratings where choice_kind_id = 2 AND choice_oboslete = False
   Else if pci_indicators_id = 3
   Select
   choice_id
   choice_sequence

choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 3 AND choice_oboslete = False
Else if pci_indicators_id = 4
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 4 AND choice_oboslete = False
Else if pci_indicators_id = 5
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 5 AND choice_oboslete = False
Else if pci_indicators_id = 6
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 6 AND choice_oboslete = False
Else if pci_indicators_id = 7
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 7 AND choice_oboslete = False
Else if pci_indicators_id = 8
Select
choice_id
choice_sequence
choice_data_value
choice_label

choice_description
From d_pci_indicator_ratings where choice_kind_id = 8 AND choice_oboslete = False
Else if pci_indicators_id = 9
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 9 AND choice_oboslete = False
Else if pci_indicators_id = 10
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 10 AND choice_oboslete = False
Else if pci_indicators_id = 11
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 11 AND choice_oboslete = False
Else if pci_indicators_id = 12
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 12 AND choice_oboslete = False
Else if pci_indicators_id = 13
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 13 AND choice_oboslete =

False
Else if pci_indicators_id = 14
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_description
From d_pci_indicator_ratings where choice_kind_id = 14 AND choice_oboslete = False

#Send to output for this PCS indicator
Output pci_indicator_id, choice_id, choice_sequence, choice_data_value, choice_label, choice_description

1.3. Output
pci_indicators_id … smallint (1, 2, 3, … 14), one per request
choice_id … smallint (1, 2, 3, 4, and 5); Choice Identifier, five per request
choice_sequence … smallint (1, 2, 3, 4, and 5); Choice Sequence, one per choice_id
choice_data_value … smallint (1, 2, 3, 4, and 5); Choice Data Value, one per choice_id
choice_label … smallint (1, 2, 3, 4, and 5); Choice Label, one per choice_id
choice_description … character varying (255); Choice Description, one per choice_id

Service GRAS-16a: Get Rangeland Health Assessment Indicators (GetRHIndicators)

Purpose: Get and return a payload of the rangeland health assessment indicators to enable the requesting application to complete a rangeland health assessment.

Service Signature

Request Payload

No data is passed into the service for processing other than requesting the service to run

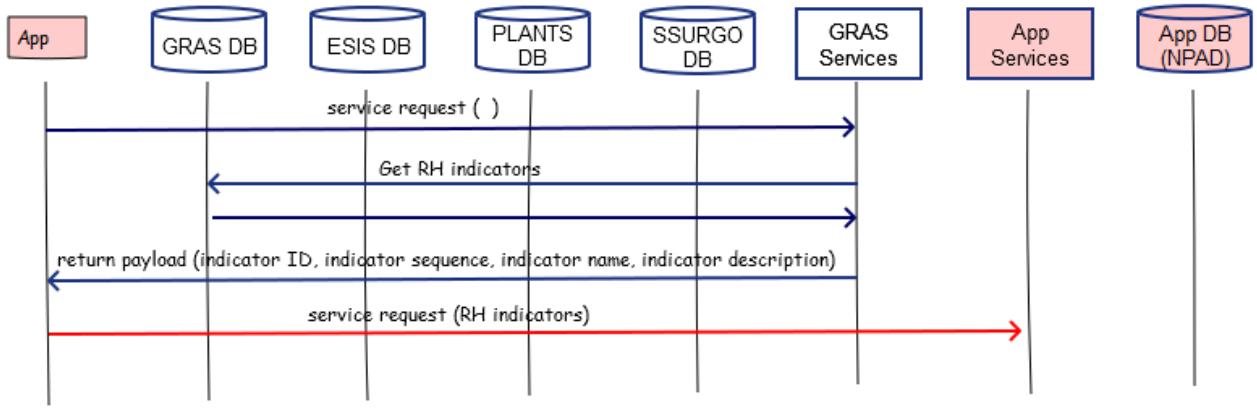
Result Payload

rhi_indicators_id … smallint (1, 2, 3, … 17), RH Indicator Identifier
rhi_indicators_sequence … smallint (1, 2, 3, … 17); Display Sequence Number for Indicator
rhi_indicators_name … character varying (75); Indicator Name
rhi_indicators_description … character varying (255); Indicator Description

Reference Data Sources

d_rhi_indicators (from GRAS data mart)

GRAS-16a: Get RH Indicators



Component

1. **Get Rangeland Health Assessment Indicators (GetRHIndicators)**
   1.1. Inputs

   1.2. Method
   Select
   rhi_indicators_id
   rhi_indicators_sequence
   rhi_indicators_name

rhi_indicators_description
From GRAS d_rhi_indicators where rhi_indicators_obsolete = False

#Send to Output
Output rhi_indicators_id, rhi_indicators_sequence, rhi_indicators_name, rhi_indicators_description

1.3. Outputs
rhi_indicators_id … smallint (1, 2, 3, … 17), RH Indicator Identifier
rhi_indicators_sequence … smallint (1, 2, 3, … 17); Display Sequence Number for RH Indicator
rhi_indicators_name … character varying (75); RH Indicator Name
rhi_indicators_description … character varying (255); RH Indicator Description

Service GRAS-16b: Get Rangeland Health Assessment Indicator Rating Choice List (GetRHIndChoiceList)

Purpose: Get and return a payload of choice lists associated with a specific rangeland health assessment indicators. User selects appropriate rating from the choice list and associated assigned numeric value score for the rating is saved to NPAD.

Example of first two indicators:

	Departure from Reference Sheet				
Indicator*	Extreme to Total	Moderate to Extreme	Moderate	Slight to Moderate	None to Slight
1. Rills _____					Reference Sheet: ____
Generic Descriptor	Rill formation is severe and well defined throughout most of the site.	Rill formation is moderately active and well defined throughout most of the site.	Active rill formation is slight at infrequent intervals; mostly in exposed areas.	No recent formation of rills; old rills have blunted or muted features.	Current or past formation of rills as expected for the site.
2. Water Flow Patterns _____					Reference Sheet: ____
Generic Descriptor	Water flow patterns extensive and numerous; unstable with active erosion; usually connected.	Water flow patterns more numerous and extensive than expected; deposition and cut areas common; occasionally connected.	Number and length of water flow patterns nearly match what is expected for the site; erosion is minor with some instability and deposition.	Number and length of water flow patterns match what is expected for the site; some evidence of minor erosion. Flow patterns are stable and short.	Matches what is expected for the site; minimal evidence of past or current soil deposition or erosion.

Service Signature

Request Payload

rhi_indicators_id … smallint (1, 2, 3, … 17); RH Indicator Identifier, one per request

Result Payload

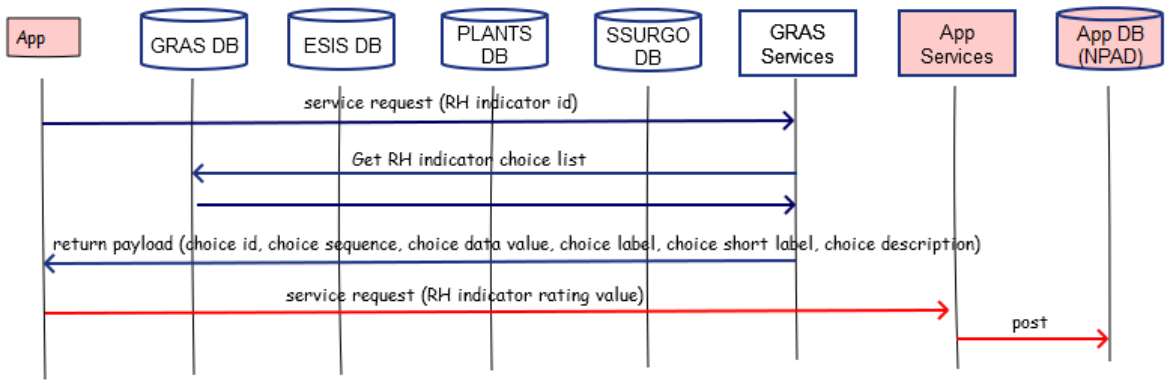
rhi_indicators_id … smallint (1, 2, 3, … 17); RH Indicator Identifier , one per request
choice_id … smallint (1, 2, 3, 4, and 5); Choice Identifier, five per request;
choice_sequence … smallint (1, 2, 3, 4, and 5); Choice Sequence, one per choice_id
choice_data_value … smallint (1, 2, 3, 4, and 5); Choice Data Value, one_per

choice_id
choice_label … character varying (50); Choice Label, one per choice_id
choice_short_label … character varying (5); Choice Short label, one per choice_id
choice_description … character varying (255); Choice Description, one per choice_id

Reference Data Sources

d_rhi_indicator_ratings (from GRAS data mart)

GRAS-16b: Get RH Indicator Rating Choice List



Component

1. **Get RH Indicator Rating Choice List (GetRHIndChoiceList)**
   1.1. Inputs
   rhi_indicators_id

   1.2. Methods
   If rhi_indicators_id = 1
   Select
   choice_id
   choice_sequence
   choice_data_value
   choice_label
   choice_short_label
   choice_description
   From d_rhi_indicator_ratings where choice_kind_id = 1 AND choice_oboslete = False
   Else if rhi_indicators_id = 2
   Select
   choice_id
   choice_sequence
   choice_data_value
   choice_label
   choice_short_label
   choice_description
   From d_rhi_indicator_ratings where choice_kind_id = 2 AND choice_oboslete =

False
Else if rhi_indicators_id = 3
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 3 AND choice_oboslete = False
Else if rhi_indicators_id = 4
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 4 AND choice_oboslete = False
Else if rhi_indicators_id = 5
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 5 AND choice_oboslete = False
Else if rhi_indicators_id = 6
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 6 AND choice_oboslete = False
Else if rhi_indicators_id = 7
Select
choice_id
choice_sequence
choice_data_value
choice_label

choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 7 AND choice_oboslete = False
Else if rhi_indicators_id = 8
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 8 AND choice_oboslete = False
Else if rhi_indicators_id = 9
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 9 AND choice_oboslete = False
Else if rhi_indicators_id = 10
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 10 AND choice_oboslete = False
Else if rhi_indicators_id = 11
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 11 AND choice_oboslete = False
Else if rhi_indicators_id = 12
Select
choice_id

choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 12 AND choice_oboslete = False
Else if rhi_indicators_id = 13
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 13 AND choice_oboslete = False
Else if rhi_indicators_id = 14
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 14 AND choice_oboslete = False
Else if rhi_indicators_id = 15
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 15 AND choice_oboslete = False
Else if rhi_indicators_id = 16
Select
choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description
From d_rhi_indicator_ratings where choice_kind_id = 16 AND choice_oboslete = False

Else if rhi_indicators_id = 17

Select

choice_id
choice_sequence
choice_data_value
choice_label
choice_short_label
choice_description

From d_rhi_indicator_ratings where choice_kind_id = 17 AND choice_oboslete = False

#Send to output for this RH indicator

Output rhi_indicator_id, choice_id, choice_sequence, choice_data_value, choice_label, choice_short_label, choice_description

1.3. Output

rhi_indicators_id … smallint (1, 2, 3, … 17); RH Indicator Identifier, one per request
choice_id … smallint (1, 2, 3, 4, and 5); Choice Identifier, five per request
choice_sequence … smallint (1, 2, 3, 4, and 5); Choice Sequence, one per choice_id
choice_data_value … smallint (1, 2, 3, 4, and 5); Choice Data Value, one per choice_id
choice_label … character varying (50); Choice Label, one per choice_id
choice_short_label … character varying (5); Choice Short Label, one per choice_id
choice_description … character varying (255); Choice Description, one per choice_id

Service GRAS-16c: Get Rangeland Health Assessment Attributes (GetRHAttributes)

Purpose: Get and return a payload of the rangeland health assessment attributes to enable the requesting application to complete a rangeland health assessment.

Service Signature

Request Payload

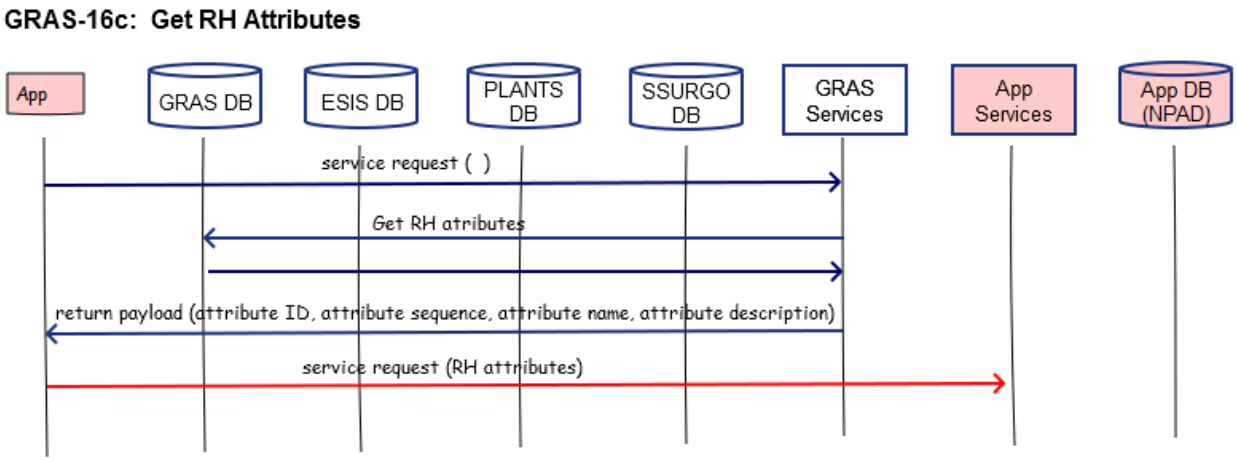
No data is passed into the service for processing other than requesting the service to run

Result Payload

rhi_attributes_id … smallint (1, 2, 3), RH Attribute Identifier
rhi_attributes_sequence … smallint (1, 2, 3); Display Sequence Number for Attribute
rhi_attributes_name … character varying (25); RH Attribute Name
rhi_attributes_description … character varying (255); RH Attribute Description

Reference Data Sources

d_rhi_attributes (from GRAS data mart)



Component

1. **Get Rangeland Health Assessment Attributes (GetRHAttributes)**
   1.1. Inputs

   1.2. Method
   Select
   rhi_attributes_id
   rhi_attributes_sequence
   rhi_attributes_name

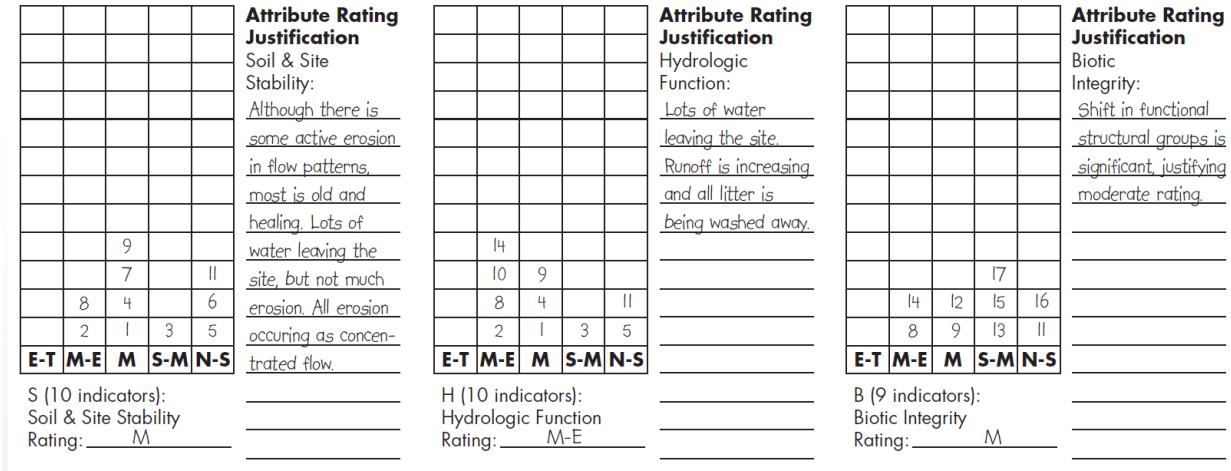
rhi_attributes_description
From GRAS d_rhi_attributes where rhi_attributes_obsolete = False

#Send to Output
Output rhi_attributes_id, rhi_attributes_sequence, rhi_attributes_name, rhi_attributes_description

1.3. Outputs
rhi_attributes_id … smallint (1, 2, 3), RH Attribute Identifier
rhi_attributes_sequence … smallint (1, 2, 3); Display Sequence Number for RH Attribute
rhi_attributes_name … character varying (25); RH Attribute Name
rhi_attributes_description … character varying (255); RH Attribute Description

Service GRAS-16d: Get Rangeland Health Assessment Attribute Rating Choice List (GetRHAttrChoiceList)

Purpose: Get and return a payload of choice lists associated with a specific rangeland health assessment attributes. User selects appropriate rating from the choice list and associated assigned numeric value score for the rating is saved to NPAD.



					Attribute Rating Justification Soil & Site Stability: Although there is some active erosion in flow patterns, most is old and healing. Lots of water leaving the site, but not much erosion. All erosion occuring as concentrated flow.
		9			
		7		11	
	8	4		6	
	2	1	3	5	
E-T	**M-E**	**M**	**S-M**	**N-S**	

S (10 indicators): Soil & Site Stability Rating: M

					Attribute Rating Justification Hydrologic Function: Lots of water leaving the site. Runoff is increasing and all litter is being washed away.
	14				
	10	9			
	8	4		11	
	2	1	3	5	
E-T	**M-E**	**M**	**S-M**	**N-S**	

H (10 indicators): Hydrologic Function Rating: M-E

					Attribute Rating Justification Biotic Integrity: Shift in functional structural groups is significant, justifying moderate rating.
			17		
	14	12	15	16	
	8	9	13	11	
E-T	**M-E**	**M**	**S-M**	**N-S**	

B (9 indicators): Biotic Integrity Rating: M

Service Signature

Request Payload

No data is passed into the service for processing other than requesting the service to run

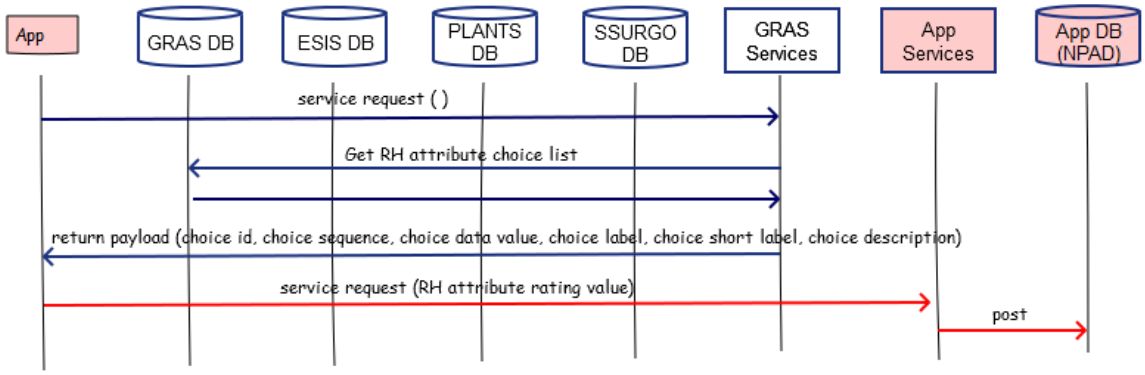
Result Payload

choice_id … smallint (1, 2, 3, 4, and 5); Choice Identifier, five per request;
choice_sequence … smallint (1, 2, 3, 4, and 5); Choice Sequence, one per choice_id
choice_data_value … smallint (1, 2, 3, 4, and 5); Choice Data Value, one_per choice_id
choice_label … character varying (25); Choice Label, one per choice_id
choice_short_label … character varying (5); Choice Short label, one per choice_id
choice_description … character varying (100); Choice Description, one per choice_id

Reference Data Sources

d_rhi_attribute_ratings (from GRAS data mart)

GRAS-16d: Get RH Attribute Rating Choice List



Component

1. **Get RH Attribute Rating Choice List (GetRHAttrChoiceList)**

   1.1. Inputs

   1.2. Methods

   Select

   choice_id
   choice_sequence
   choice_data_value
   choice_label
   choice_short_label
   choice_description

   From GRAS d_rhi_attribute_ratings where choice_oboslete = False

   #Send to output for this RH attribute

   Output choice_id, choice_sequence, choice_data_value, choice_label, choice_short_label, choice_description

   1.3. Output

   choice_id … smallint (1, 2, 3, 4, and 5); Choice Identifier, five per request
   choice_sequence … smallint (1, 2, 3, 4, and 5); Choice Sequence, one per choice_id
   choice_data_value … smallint (1, 2, 3, 4, and 5); Choice Data Value, one per choice_id
   choice_label … character varying (25); Choice Label, one per choice_id
   choice_short_label … character varying (5); Choice Short Label, one per choice_id
   choice_description … character varying (100); Choice Description, one per choice_id

Service GRAS-16e: Get Rangeland Health Assessment Ecological Site Reference Data (GetRHESDRefData)

Purpose: Get and return a payload of ESD reference data to be utilized in verifying that the rangeland health assessment is being conducted on the correct ecological site. User collects actual data from the evaluation area and compares to the reference data to verify that correct ecological site is being assessed.

Excerpt below from Interpreting Indicators of Rangeland Health, Version 4 technical reference showing reference data for site verification. User selects either ESD or Soil Survey methods to assemble site verification reference data. This service specification returns reference data when ESD method selected.

ESDs do not contain diagnostic horizon information or surface effervescence as data elements. Information could be contained in text narratives within ESD. Therefore, not included in return payload.

Soil depth will be presented as minimum and maximum depths in inches.

Soil/site verification:
Range/Ecol. Site Descr., Soil Surv., and/or Ecol. Ref. Area:
Surface texture grfsl, grlfs, gl
Depth: very shallow __, shallow __, moderate __, deep X
Type and depth of diagnostic horizons:
1. Calcic horizon w/in 20" 3. _____
2. _____ 4. _____
Surf. Efferv.: none __, v. slight __, slight __, strong X, violent __

Parent material Alluvium Slope 0-5 % Elevation 4100 ft.

Average annual precipitation 8-12 inches

Service Signature

Request Payload

es_id … character varying(60); one per request; Ecological Site Identifier

Result Payload

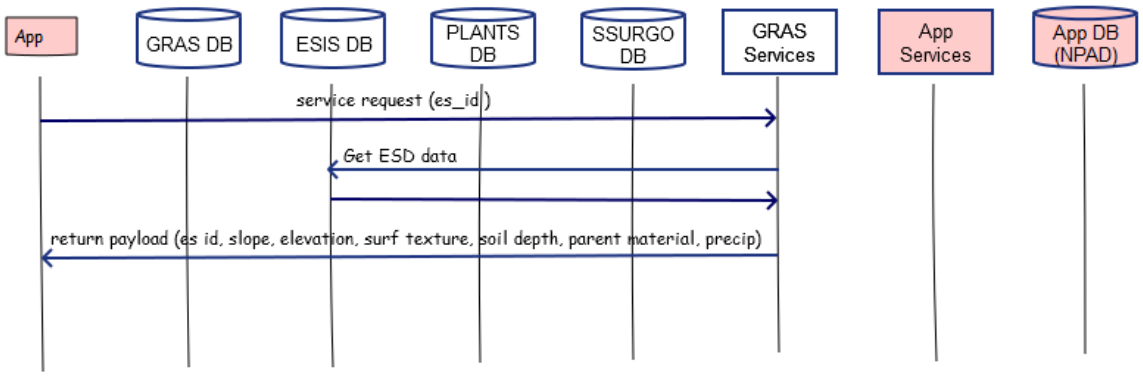
es_id… character varying(60); one per request; Ecological Site Identifier
slope_gradient_min … numeric(3,0); Slope Minimum Percent
slope_gradient_max … numeric(3,0); Slope Maximum Percent
elevation_min … numeric(5,0); Elevation Minimum Feet

elevation_max … numeric(5,0); Elevation Maximum Feet
ref_surface_tex1 … character varying(60); Surface Texture 1
ref_surface_tex2 … character varying(60); Surface Texture 2
ref_surface_tex3 … character varying(60); Surface Texture 3
soil_depth_min … numeric(4,0); Soil Depth Minimum Inches
soil_depth_max … numeric(4,0); Soil Depth Maximum Inches
ref_parent_mtrl_kind1 … character varying(80); Parent Material Kind 1
ref_parent_mtrl_kind2 … character varying(80); Parent Material Kind 2
ref_parent_mtrl_kind3 … character varying(80); Parent Material Kind 3
mean_annual_precip_min … numeric(5,2); Mean Annual Precipitation Minimum Inches
mean_annual_precip_max … numeric(5,2); Mean Annual Precipitation Maximum Inches

Reference Data Sources

Ecological Site Information System (ESIS) Data Mart
- parent_material_kind_lkp table
- physiographic_descriptions table
- rep_climate_features table
- rep_soil_features table
- surface_texture_class_lkp table
- surface_texture_mod_lkp table

GRAS-16e: Get RH Ecological Site Reference Data



Component

1. **Get RH Ecological Site Reference Data (GetRHESDRefData)**
   1.1. Inputs
   es_id

   1.2. Data
   ESIS
   parent_material_kind_lkp

parent_material_kind_id
parent_material_kind_label
physiographic_descriptions
es_type
es_mlra
es_mlru
es_site_number
es_state
slope_gradient_min
slope_gradient_max
elevation_min
elevation_max
rep_climate_features
es_type
es_mlra
es_mlru
es_site_number
es_state
mean_annual_precip_min
mean_annual_precip_max
rep_soil_features
es_type
es_mlra
es_mlru
es_site_number
es_state
parent_material_kind1
parent_material_kind2
parent_material_kind3
surface_texture_class1
surface_texture_modifier1
surface_texture_class2
surface_texture_modifier2
surface_texture_class3
surface_texture_modifier3
soil_depth_min
soil_depth_max
surface_texture_class_lkp
texture_class_id
texture_class_choice
surface_texture_mod_lkp
texture_mod_id
texture_mod_choice

1.3. Methods

For es_id

#Get parent material kind1, 2, and 3; surface texture class1, 2, and 3; surface

texture modifier1, 2, and 3; soil depth min and max for ecological site from ESIS rep_soil_features table
Select
parent_material_kind1
parent_material_kind2
parent_material_kind3
surface_texture_class1
surface_texture_modifier1
surface_texture_class2
surface_texture_modifier2
surface_texture_class3
surface_texture_modifier3
soil_depth_min
soil_depth_max
Into ref_soil_feat table
From ESIS rep_soil_features table
Where concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=es_id

#Get concatenated surface texture1 and surface texture modifier1 label for ecological site
Select
surface_texture_class1
surface_texture_modifier1
surface_texture_class_lkp.texture_class_choice
surface_texture_mod_lkp.texture_mod_choice
Into ref_surface_texture1 table
From ref_soil_feat table
Left Join ESIS surface_texture_class_lkp table
On
ref_soil_feat.surface_texture_class1=surface_texture_class_lkp.texture_class_id
Left Join ESIS surface_texture_mod_lkp table
On
ref_soil_feat.surface_texture_modifier1=surface_texture_mod_lkp.texture_mod_id

Alter Table ref_surface_texture1
Add concatenated(texture_mod_choice, texture_class_choice) as ref_surface_tex1

#Get concatenated surface texture2 and surface texture modifier2 label for ecological site
Select
surface_texture_class2
surface_texture_modifier2
surface_texture_class_lkp.texture_class_choice
surface_texture_mod_lkp.texture_mod_choice
Into ref_surface_texture2 table

From ref_soil_feat table
Left Join ESIS surface_texture_class_lkp table
On
ref_soil_feat.surface_texture_class2=surface_texture_class_lkp.texture_class_id
Left Join ESIS surface_texture_mod_lkp table
On
ref_soil_feat.surface_texture_modifier2=surface_texture_mod_lkp.texture_mod_id

Alter Table ref_surface_texture2
Add concatenated(texture_mod_choice, texture_class_choice) as ref_surface_tex2

#Get concatenated surface texture3 and surface texture modifier3 label for ecological site
Select
surface_texture_class3
surface_texture_modifier3
surface_texture_class_lkp.texture_class_choice
surface_texture_mod_lkp.texture_mod_choice
Into ref_surface_texture3 table
From ref_soil_feat table
Left Join ESIS surface_texture_class_lkp table
On
ref_soil_features.surface_texture_class3=surface_texture_class_lkp.texture_class_id
Left Join ESIS surface_texture_mod_lkp table
On
ref_soil_features.surface_texture_modifier3=surface_texture_mod_lkp.texture_mod_id

Alter Table ref_surface_texture3
Add concatenated(texture_mod_choice, texture_class_choice) as ref_surface_tex3

#Get parent material kind1 label for ecological site
Select
parent_material_kind1
parent_material_kind_lkp.material_kind_label
Into ref_parent_material_kind1 table
From ref_soil_feat table
Inner Join ESIS parent_material_kind_lkp table
On
ref_soil_feat.parent_material_kind1=parent_material_kind_lkp.parent_material_kind_id

Alter Table ref_ parent_material_kind1

Add material_kind_label as ref_parent_mtrl_kind1

#Get parent material kind2 label for ecological site
Select
parent_material_kind2
parent_material_kind_lkp.material_kind_label
Into ref_parent_material_kind2 table
From ref_soil_feat table
Inner Join ESIS parent_material_kind_lkp table
On
ref_soil_feat.parent_material_kind2=parent_material_kind_lkp.parent_material_kind_id

Alter Table ref_ parent_material_kind2
Add material_kind_label as ref_parent_mtrl_kind2

#Get parent material kind3 label for ecological site
Select
parent_material_kind3
parent_material_kind_lkp.material_kind_label
Into ref_parent_material_kind3 table
From ref_soil_feat table
Inner Join ESIS parent_material_kind_lkp table
On
ref_soil_feat.parent_material_kind3=parent_material_kind_lkp.parent_material_kind_id

Alter Table ref_ parent_material_kind3
Add material_kind_label as ref_parent_mtrl_kind3

#Get mean annual precipitation minimum and maximum for ecological site
Select
mean_annual_precip_min
mean_annual_precip_max
Into ref_mean_annual_precip table
From ESIS rep_climate_features
Where concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=es_id

#Get aspect, slope minimum and maximum, and elevation minimum and maximum for ecological site
Select
slope_gradient_min
slope_gradient_max
elevation_min
elevation_max
Into ref_physiographic_desc table
From ESIS rep_physiographic_description

Where concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=es_id

#Get aspect, slope minimum and maximum, elevation minimum and maximum, surface texture1, surface texture2, surface texture3, parent material kind1, parent material kind2, parent material kind3, and mean annual precipitation minimum and maximum for ecological site and combine into one table
Select
ref_physiographic_desc.slope_gradient_min
ref_physiographic_desc.slope_gradient_max
ref_physiographic_desc.elevation_min
ref_physiographic_desc.elevation_max
ref_surface_texture1.ref_surface_tex1
ref_surface_texture2.ref_surface_tex2
ref_surface_texture2.ref_surface_tex2
ref_surface_texture2.soil_depth_min
ref_surface_texture2.soil_depth_max
ref_parent_material_kind1.ref_parent_mtrl_kind1
ref_parent_material_kind2. ref_parent_mtrl_kind2
ref_parent_material_kind2. ref_parent_mtrl_kind3
ref_mean_annual_precip.mean_annual_precip_min
ref_mean_annual_precip.mean_annual_precip_max

Into esd_ref_site_verification table

Resulting esd_ref_site_verification table
es_id
slope_gradient_min
slope_gradient_max
elevation_min
elevation_max
ref_surface_tex1
ref_surface_tex2
ref_surface_tex3
soil_depth_min
soil_depth_max
ref_parent_mtrl_kind1
ref_parent_mtrl_kind2
ref_parent_mtrl_kind3
mean_annual_precip_min
mean_annual_precip_max

#Send to output
Output data in esd_ref_site_verification table for this es_id

1.4. Output
es_id… character varying(60); one per request; Ecological Site Identifier
slope_gradient_min … numeric(3,0); Slope Minimum Percent

slope_gradient_max … numeric(3,0); Slope Maximum Percent
elevation_min … numeric(5,0); Elevation Minimum Feet
elevation_max … numeric(5,0); Elevation Maximum Feet
ref_surface_tex1 … character varying(60); Surface Texture 1
ref_surface_tex2 … character varying(60); Surface Texture 2
ref_surface_tex3 … character varying(60); Surface Texture 3
soil_depth_min … numeric(4,0); Soil Depth Minimum Inches
soil_depth_max … numeric(4,0); Soil Depth Maximum Inches
ref_parent_mtrl_kind1 … character varying(80); Parent Material Kind 1
ref_parent_mtrl_kind2 … character varying(80); Parent Material Kind 2
ref_parent_mtrl_kind3 … character varying(80); Parent Material Kind 3
mean_annual_precip_min … numeric(5,2); Mean Annual Precipitation Minimum Inches
mean_annual_precip_max … numeric(5,2); Mean Annual Precipitation Maximum Inches

Service GRAS-16f: Get Rangeland Health Assessment SSURGO Reference Data (GetRHSSURGORefData)

Purpose: Get and return a payload of SSURGO reference data to be utilized in verifying that the rangeland health assessment is being conducted on the correct ecological site. User collects actual data from the evaluation area and compares to the reference data to verify that correct ecological site is being assessed.

Excerpt below from Interpreting Indicators of Rangeland Health, Version 4 technical reference showing reference data for site verification. User selects either ESD or Soil Survey methods to assemble site verification reference data. This service specification returns reference data when SSURGO method selected.

Soil/site verification:
Range/Ecol. Site Descr., Soil Surv., and/or Ecol. Ref. Area:
Surface texture grfsl, grlfs, gl
Depth: very shallow __, shallow __, moderate __, deep X
Type and depth of diagnostic horizons:
1. Calcic horizon w/in 20" 3. _____
2. _____ 4. _____
Surf. Efferv.: none __, v. slight __, slight __, strong X, violent __

Parent material Alluvium Slope 0-5 % Elevation 4100 ft.

Average annual precipitation 8-12 inches

Service Signature

Request Payload

cokey … character varying(60); one per request; Soil Component Key

Result Payload

cokey … character varying(60); one per request; Soil Component Key
slope_l … numeric(3,0); Slope Minimum Percent
slope_h … numeric(3,0); Slope Maximum Percent
elevation_ft_l … numeric(5,0); Elevation Minimum Feet
elevation_ft_h … numeric(5,0); Elevation Maximum Feet
map_in_l … numeric(5,2); Mean Annual Precipitation Minimum Inches
map_in_h … numeric(5,2); Mean Annual Precipitation Maximum Inches
soil_depth_in … numeric(4,0); Soil Depth Maximum Inches

chkey … character varying(30); Soil Component Horizon Key
texture … character varying(30); one or more; Texture Modifier and Class

rvindicator … Boolean(3); Relative Value Indicator

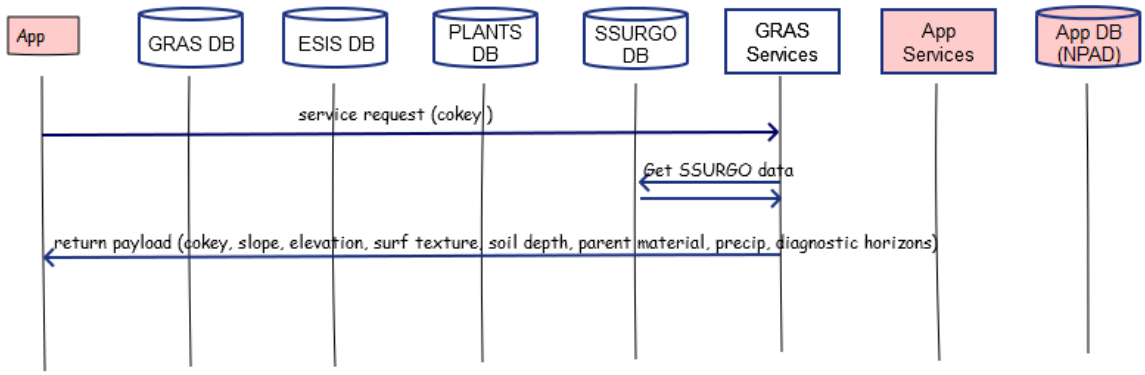
copmgrpkey … character varying(30); Soil Component Parent Material Group Key
pmkind … character varying(254); one or more; Soil Component Parent Material Kind
pmmodifier … character varying(254); one per pmkind; Soil Component Parent Material Kind Modifier

codiagfeatkey … character varying(30); Soil Component Diagnostic Features Key
featkind … character varying(254); one or more; Soil Component Diagnostic Feature Kind
featdept_in_r … numeric(4,0); Soil Component Diagnostic Feature Relative Value Top Depth Inches
featdepb_in_r … numeric(4,0); Soil Component Diagnostic Feature Relative Value Bottom Depth Inches

Reference Data Sources

SSURGO Data Mart
component table
chorizon table
chtexturegrp table
copmgrp table
copm table
codiagfeatures table

GRAS-16f: Get RH SSURGO Reference Data



Component

1. **Get Rangeland Health SSURGO Reference Data (GetRHSSURGORefData)**
   1.1. Inputs
   cokey

   1.2. Data
   SSURGO

component
- cokey
- slope_l
- slope_h
- elev_l
- elev_h
- map_l
- map_h

chorizon
- cokey
- chkey
- hzdepb_r

chtexturegrp
- chkey
- texture
- rvindicator

copmgrp
- cokey
- copmgrpkey

copm
- copmgrpkey
- pmkind
- pmmodifier

codiagfeatures
- cokey
- featkind
- featdept_r
- featdepb_r

1.3. Methods

For cokey

#Get slope min and max; elevation min and max; mean annual precipitation min and max for soil component from SSURGO component table. Convert elevation (meters to feet) and mean annual precipitation (millimeters to inches) values from metric to English units.

Select
- slope_l
- slope_h
- elev_l
- elev_h
- map_l
- map_h

Into ref_soil_comp table
From component table
Where cokey=component.cokey

Alter Table ref_soil_comp

Add (elev_l * 3.28084) as elev_ft_l
Add (elev_h * 3.28084) as elev_ft_h
Add (map_l * 0.03937) as map_in_l
Add (map_h * 0.03937) as map_in_h

Resulting ref_soil_comp table
    slope_l
    slope_h
    elev_l
    elev_h
    elev_ft_l
    elev_ft_h
    map_l
    map_h
    map_in_l
    map_in_h

#Get texture depth for soil component chorizon table. Utilize the horizon bottom depth relative value for the lowest horizon. Convert soil depth (centimeters to inches) value from metric to English units.
Select
    chorizon.hzdepb_r
Into ref_soil_horizons_depth table
From component table
Inner Join chorizon table
On component.cokey=chorizon.cokey

Select
    Max(chorizon.hzdepb_r) as soil_depth
Into ref_soil_depth table
From ref_soil_horizons_depth table

Alter Table ref_soil_comp
Add (soil_depth * 0.3937) as soil_depth_in

Resulting ref_soil_comp table
    slope_l
    slope_h
    elev_l
    elev_h
    elev_ft_l
    elev_ft_h
    map_l
    map_h
    map_in_l
    map_in_h
    soil_depth_in

#Send to output
Output slope_l, slope_h, elev_ft_l, elev_ft_h, map_in_l, map_in_h, and soil_depth_in for this cokey

#Get surface soil texture where horizon to depth relative value equals zero from chorizon table joined with chtexturegrp table. There can be one or more surface textures values for a soil component. The relative value indicator (Yes or No) is also returned for each assigned surface texture value.
Select
chorizon.chkey
chtexturegrp.texture
chtexturegrp.rvindicator
Into ref_surface_texture table
From component table
Inner Join chorizon table
(chorizon table **Inner Join** chtexturegrp table
On chorizon.chkey=chtexturegrp.chkey)
On component.cokey=chorizon.cokey
Where chorizon.hzdept_r = 0

Resulting ref_surface_texture table
chkey
texture
rvindicator

#Send to output
Output data in ref_surface_texture table for this cokey

#Get parent material kinds for the soil component from the copmgrp table joined with copm table. There can be one or more parent material kinds for a soil component.
Select
copmgrp.copmgrpkey
copm.pmkind
copm.pmmodifier
Into ref_pm_kind table
From component table
Inner Join copmgrp table
(copmgrp table **Inner Join** copm table
On copmgrp.copmgrpkey=copm.copmgrpkey)
On component.cokey=copmgrp.cokey

Resulting ref_pm_kind table
copmgrpkey
pmkind
pmmodifier

#Send to output
Output data in esd_ref_ pm_kind table for this cokey

#Get diagnostic horizons and top and bottom height relative values for each horizon for the soil component from codiagfeatures table. There can be one or more diagnostic horizons for the soil component. Convert diagnostic horizon depth (centimeters to inches) values from metric to English units.
Select
codiagfeatures.codiagfeatkey
codiagfeatures.featkind
codiagfeatures.featdept_r
codiagfeatures.featdepb_r
Into ref_diag_horizons table
From component table
Inner Join codiagfeatures table
On component.cokey=codiagfeatures.cokey

Alter Table ref_diag_horizons
Add (featdept_r * 0.3937) as featdept_in_r
Add (featdepb_r * 0.3937) as featdepb_in_r

Resulting ref_diag_horizons table
codiagfeatkey
featkind
featdept_r
featdepb_r
featdept_in_r
featdepb_in_r

#Send to output
Output data in esd_ref_ diag_horizons table for this cokey

1.4. Output

cokey … character varying(60); one per request; Soil Component Key
slope_l … numeric(3,0); Slope Minimum Percent
slope_h … numeric(3,0); Slope Maximum Percent
elevation_ft_l … numeric(5,0); Elevation Minimum Feet
elevation_ft_h … numeric(5,0); Elevation Maximum Feet
map_in_l … numeric(5,2); Mean Annual Precipitation Minimum Inches
map_in_h … numeric(5,2); Mean Annual Precipitation Maximum Inches
soil_depth_in … numeric(4,0); Soil Depth Maximum Inches

chkey … character varying(30); Soil Component Horizon Key
texture … character varying(30); one or more; Texture Modifier and Class
rvindicator … Boolean(3); Relative Value Indicator

copmgrpkey … character varying(30); Soil Component Parent Material Group Key
pmkind … character varying(254); one or more; Soil Component Parent Material Kind
pmmodifier … character varying(254); one per pmkind; Soil Component Parent Material Kind Modifier

codiagfeatkey … character varying(30); Soil Component Diagnostic Features Key
featkind … character varying(254); one or more; Soil Component Diagnostic Feature Kind
featdept_in_r … numeric(4,0); Soil Component Diagnostic Feature Relative Value Top Depth Inches
featdepb_in_r … numeric(4,0); Soil Component Diagnostic Feature Relative Value Bottom Depth Inches

Service GRAS-16g: Get Soil Component and Ecological Site for Rangeland Health (RH) Assessment Evaluation Area (GetRHCompESD)

Purpose: Get pasture number, state and field office for office submitting request, map unit, soil component and ecological site information for rangeland health assessment evaluation area. The information is utilized to populate front page of Evaluation Sheet. User selects the soil component and assigned ecological site from choice generated for the map unit. When user selects either ESD or Soil Survey methods to assemble RH evaluation site verification reference data, the soil component (cokey) or ecological site (es_id) of the soil component/ecological that was identified for the evaluation area is utilized as the request payload for either GRAS-16e or GRAS-16f.

Excerpt below from Interpreting Indicators of Rangeland Health, Version 4 technical reference.

Evaluation Sheet (Example) (Front)

Aerial Photo:_____

Management Unit: Allotment 1, pasture 1 State: NM Office: Las Cruces Range/Ecol. Site Code: 042XB999NM
(Allotment or pasture)

Ecological Site Name: Limy Soil Map Unit/Component Name: Nickel gravelly fine sandy loam

Observers: Joe Smith, Jose Garcia, and Thaddeus Jones Date: June 10, 2002

Location (description): Limy site two miles north of windmill in S.E. pasture

T. 11 S R. 23 W or _____N. Lat. Or UTM E_____m Position by GPS? Y / N No
UTM Zone___, Datum___

Sec. 12, NE 1/4 _____W. Long. N_____m Photos taken? Y / N Yes

Size of evaluation area: Evaluation area is approximately 3 ac. and represents entire ecological site in this pasture

Service Signature

Request Payload

AoAID … integer; one per request payload; Area of Analysis (AoA) Identifier
servicing_office_id … integer; Office ID of office submitting request
servicing_office_state_county_code … character(5); State county code of office submitting request payload
site_id … integer; one per request payload; Evaluation Site Identifier
rh_site_geometry … point; Rangeland Health Assessment Site Geometry

Result Payload

AoAID … integer; Area of Analysis (AoA) Identifier
servicing_office_id … integer; Office ID of office submitting request
servicing_office_state_county_code … character(5); State county code of office submitting request payload
site_id … integer; one per request payload; Evaluation Site Identifier
musym … character varying(6); Mapunit Symbol
muname … character varying(175); Mapunit Name
cokey… character varying(60), one or more in the FIS, Soil Component Key
compname … character varying(60); Soil Component Name

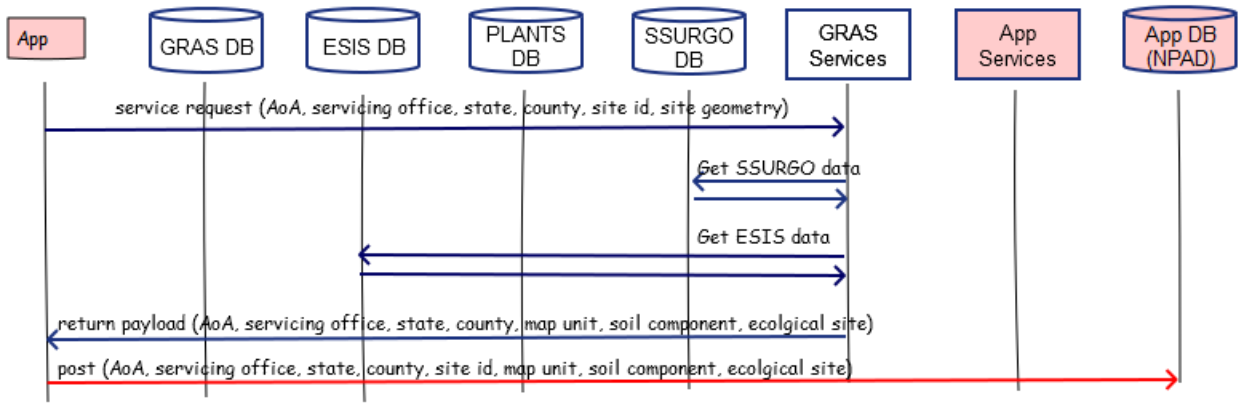
comppct_r … smallint; Soil Component Percentage Relative Value
es_id … character varying(60), one or more per FIS, Ecological Site Identifier
es_range_name … character varying(120), Ecological Site Name, name applied if ecological site identifier begins with R

Reference Data Sources

SSURGO Data Mart
- mapunit table
- component table
- coecoclass table

ESIS Data Mart
- Ecological_Sites table

GRAS-16g: Get Soil Component and ES for RH Assessment Area



Component

1. **Get Soil Component and Ecological Site for Rangeland Health (RH) Assessment Evaluation Area (GetRHCompESD)**

1.1. Inputs

AoAId	1
servicing_office_id	46620
servicing_office_state_county_code	48377
site_id	10
rh_site_geometry	30.2180881043899, -103.979251725913

1.2. Data
SSURGO
- mapunit
  - mukey
  - musym
  - muname
- component

cokey
compname
comppct_r
coecoclass
cokey
ecoclassid
ESIS
Ecological_Sites table
es_type
es_mlra
es_mlru
es_site_number
es_state
range_site_primary_name
range_site_secondary_name
range_site_tertiary_name

1.3. GIS Operations
For AoAId
For site_id
#Get mapunit associated with rangeland health evaluation site.
Intersect RH site point geometry with SSURGO mapunit geometry

rh_mu_attrib table columns
AoAId
site_id
mu_key

1.4. Methods
For AoAId
For site_id
#Create RH mapunit component table
Select
rh_mu_attrib.AoAId
rh_mu_attrib.site_id
mapunit.mu_key
mapunit.musym
mapunit.muname
component.cokey
component.compname
component.comppct_r
coecoclass.coecoclassid

Into rh_mu_comp attrib table
From rh_mu_attrib table
Inner Join mapunit table
(mapunit table **Inner Join** component table
On mapunit.mukey=component.mukey)

(component table **Left Join** coecoclass table
On component.cokey=coecoclass.cokey)
On rh_mu_attrib.mukey=mapunit.mukey

Resulting rh_mu_comp attrib table
AoAId
site_id
mukey
musym
muname
cokey
compname
comppct_r
ecoclassid

#Create RH mapunit component ecological site table
Select
rh_mu_comp_attrib.AoAId
rh_mu_comp_attrib.site_id
rh_mu_comp_attrib.mu_key
rh_mu_comp_attrib.musym
rh_mu_comp_attrib.muname
rh_mu_comp_attrib.cokey
rh_mu_comp_attrib.compname
rh_mu_comp_attrib.comppct_r
ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state) **As** es_id
ecological_sites.concatenated(range_site_primary_name, range_site_secondary_name, range_site_tertiary_name) **As** es_range_name

Into rh_mu_comp__es attrib table
From rh_mu_comp attrib table
Left Join ecological_sites table
On rh_mu_comp.ecoclassid=ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)
Where es_id LIKE 'R%'
Order by rh_mu_comp.comppct_r Descending

Resulting rh_mu_comp attrib table
AoAId
site_id
mukey
musym
muname
cokey
compname

comppct_r
es_id
es_range_name

#Send to output
Output data in rh_mu_comp_es table for this AoAId and site_id

#The data in rh_mu_comp_es should enable the application to create the following choice lists to allow use to select desired soil component and assigned ecological site.

Map Unit Symbol	Map Unit Name	Component Name	Component %	Ecological Site ID	Ecological Site Name	Select
musym	Muname	compname	comppct_r	es_id	es_range_name	
STE	Strawhouse-Stillwell complex, 1 to 30 percent slopes	Strawhouse	50	R042XG735TX	Gravelly, Hot Desert Shrub	☐
		Stillwell	35	R042XG735TX	Gravelly, Hot Desert Shrub	☐
		Unnamed	10	NA	NA	☐
		Geefour	5	NA	NA	☐

#Pass servicing office and servicing office state and county code for this AoAId from Input to Output
Output servicing_office_id, servicing_office_state_county_code

1.5. Output

AoAID … integer; Area of Analysis (AoA) Identifier
servicing_office_id … integer; Office ID of office submitting request
servicing_office_state_county_code … character(5); State county code of office submitting request payload
site_id … integer; one per request payload; Evaluation Site Identifier
musym … character varying(6); Mapunit Symbol
muname … character varying(175); Mapunit Name
cokey… character varying(60), one or more in the FIS, Soil Component Key
compname … character varying(60); Soil Component Name
comppct_r … smallint; Soil Component Percentage Relative Value
es_id … character varying(60), one or more per FIS, Ecological Site Identifier
es_range_name … character varying(120), Ecological Site Name, name applied if ecological site identifier begins with R

Service GRAS-16h: Get Rangeland Health Assessment SSURGO Choice Lists (GetRHSSURGOChoiceLists)

Purpose: Get and return a payload of choice lists from SSURGO domain tables to be utilized in populating surface texture, type of diagnostic horizons, and topographic position for the rangeland health assessment evaluation area.

Excerpt below from Interpreting Indicators of Rangeland Health, Version 4 technical reference showing data collected for the evaluation area.

Evaluation Area:
Surface texture gfsl
Depth: very shallow __, shallow __, moderate __, deep X
Type and depth of diagnostic horizons:
1. Calcic horizon at 15" 3. _____
2. _____ 4. _____
Surf. Efferv.: none __, v. slight __, slight __, strong X, violent __

Topographic position toeslope Aspect south

Seasonal distribution Summer thunderstorms dominate

Service Signature

Request Payload

No data is passed into the service for processing other than requesting the service to run

Result Payload

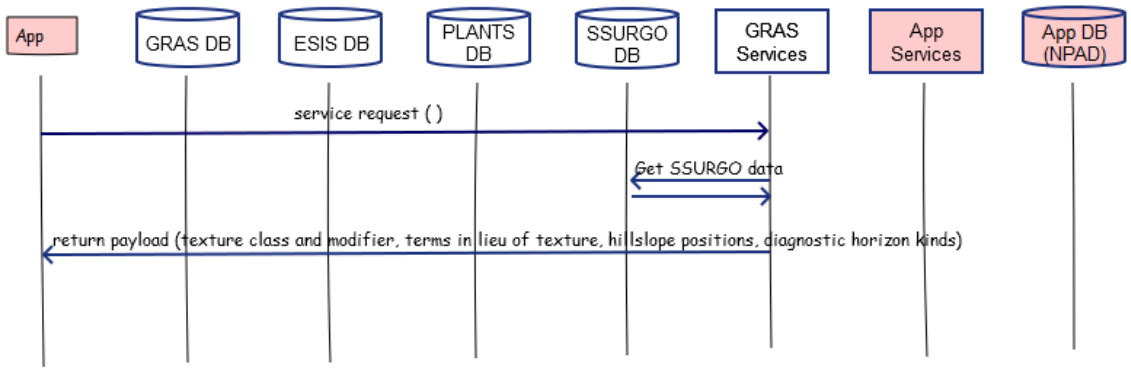
choice_kind_id … smallint; 1, 2, 3, 4, and 5; Choice Kind Identifier
choice_kind … character varying(20); one per choice_kind_id; Choice Kind
Seq … smallint; Choice Sequence
Choice ID … smallint; Choice Identifier
Choice Data Text Entry … character varying(50); Choice Data Text Entry
Choice Label … character varying(50); Choice Label
Choice Description … character varying; Choice Description

Reference Data Sources

SSURGO Domain Tables
diag_horz_feat_kind table
hillslope_profile table

terms_used_in_lieu_of_texture table
texture_class table
texture_modifier table

GRAS-16h: Get RH SSURGO Choice Lists



Component

1. **Get Rangeland Health SSURGO Choice Lists (GetRHSSURGOChoiceLists)**
   1.1. Inputs
   None

   1.2. Data
   SSURGO domain tables
   diag_horz_feat_kind
   Seq
   Obsolete?
   Choice ID
   Choice Data Text Entry
   Choice Label
   Choice Description
   hillslope_profile
   Seq
   Obsolete?
   Choice ID
   Choice Data Text Entry
   Choice Label
   Choice Description
   texture_class
   Seq
   Obsolete?
   Choice ID
   Choice Data Text Entry
   Choice Label
   Choice Description

texture_modifier
- Seq
- Obsolete?
- Choice ID
- Choice Data Text Entry
- Choice Label
- Choice Description

terms_used_in_lieu_of_texture
- Seq
- Obsolete?
- Choice ID
- Choice Data Text Entry
- Choice Label
- Choice Description

1.3. Methods

#Get texture classes from SSURGO texture_class domain table to populate choice list when selecting texture class for surface texture.

Select
- Seq
- Choice ID
- Choice Data Text Entry
- Choice Label
- Choice Description

Into rh_ssurgo_choice_list table
From texture_class table
Where Obsolete? = No

Alter Table rh_ssurgo_choice_list
Add textclass as choice_kind
Add 1 as choice_kind_id

Resulting rh_ssurgo_choice_list table
- choice_kind_id=1
- choice_kind=textclass
- Seq
- Choice ID
- Choice Data Text Entry
- Choice Label
- Choice Description

 # Get texture modifiers from SSURGO texture_modifier domain table to populate choice list when selecting a texture modifier for surface texture.

Select
- Seq
- Choice ID
- Choice Data Text Entry

Choice Label
Choice Description
Into rh_ssurgo_choice_list table
From texture_modifier table
Where Obsolete? = No

Alter Table rh_ssurgo_choice_list
Add textmod as choice_kind
Add 2 as choice_kind_id

Resulting rh_ssurgo_choice_list table
choice_kind_id=1 and 2
choice_kind=textclass and textmod
Seq
Choice ID
Choice Data Text Entry
Choice Label
Choice Description

#Get terms in lieu of texture from SSURGO terms_used_in_lieu_of_texture domain table to populate choice list when selecting a term in lieu of texture for surface texture.

Select
Seq
Choice ID
Choice Data Text Entry
Choice Label
Choice Description
Into rh_ssurgo_choice_list table
From terms_used_in_lieu_of_texture table
Where Obsolete? = No

Alter Table rh_ssurgo_choice_list
Add termlieu as choice_kind
Add 3 as choice_kind_id

Resulting rh_ssurgo_choice_list table
choice_kind_id=1, 2, and 3
choice_kind=textclass, textmod, and termlieu
Seq
Choice ID
Choice Data Text Entry
Choice Label
Choice Description

#Get diagnostic horizon feature kinds from SSURGO diag_horz_feat_kind domain table to populate choice list when selecting kind of diasagnostic horizon.

Select
Seq
Choice ID
Choice Data Text Entry
Choice Label
Choice Description
Into rh_ssurgo_choice_list table
From diag_horz_feat_kind table
Where Obsolete? = No

Alter Table rh_ssurgo_choice_list
Add diaghorz as choice_kind
Add 4 as choice_kind_id

Resulting rh_ssurgo_choice_list table
choice_kind_id=1, 2, 3, and 4
choice_kind=textclass, textmod, termlieu, and diaghorz
Seq
Choice ID
Choice Data Text Entry
Choice Label
Choice Description

#Get hillslope position kinds from SSURGO hillslope_profile domain table to populate choice list when selecting topographic position.
Select
Seq
Choice ID
Choice Data Text Entry
Choice Label
Choice Description
Into rh_ssurgo_choice_list table
From hillslope_profile table
Where Obsolete? = No

Alter Table rh_ssurgo_choice_list
Add hillslope as choice_kind
Add 5 as choice_kind_id

Resulting rh_ssurgo_choice_list table
choice_kind_id=1, 2, 3, 4, and 5
choice_kind=textclass, textmod, termlieu, diaghorz, and hillslope
Seq
Choice ID
Choice Data Text Entry
Choice Label
Choice Description

#Send to output

Output data in rh_ssurgo_choice_list table

#The data in rh_ssurgo_choice_list should enable the application to create the following table.

choice_ kind_ id	choice_ kind	Seq	Choice ID	Choice Data Text Entry	Choice Label	Choice Description
1	textclass	1	21	c	Clay	
1	textclass	2	17	cl	Clay loam	
1	textclass	3	1	cos	Coarse sand	
...						
2	textmod	1	71	artartv	Artifactual	15 to 35 percent human artifacts, by volume
2	textmod	2	72	artv	Very artifactual	35 to 60 percent human artifacts, by volume
2	textmod	3	73	artx	Extremely artifactual	60 to 90 percent human artifacts, by volume
2	textmod	4	26	ashy	Ashy	Ashy
…						
3	termlieu	1	2	apum	Ashy-pumiceous	Ashy-pumiceous
3	termlieu	2	56	art	Artifacts	Dominated by human artifacts with too little fine-earth to determine the textural class (less than about 10 percent fine-earth, by volume)
3	termlieu	3	1	ashy	Ashy	Ashy
…						
4	diaghorz	1	2	abrupt textural change	Abrupt textural change	
4	diaghorz	2	29	agric horizon	Agric horizon	
4	diaghorz	3	28	albic horizon	Albic horizon	
…						
5	hillslope	1	1	summit	Summit	The topographically highest hillslope position of a hillslope and exhibiting a nearly level (planar or only slightly convex) surface.
5	hillslope	2	2	shoulder	Shoulder	The hillslope position that forms the uppermost inclined surface near the top of a hillslope. If present, it comprises the transition zone from backslope to summit. The surface id sominantly convex in profile and erosional in origin.
…						

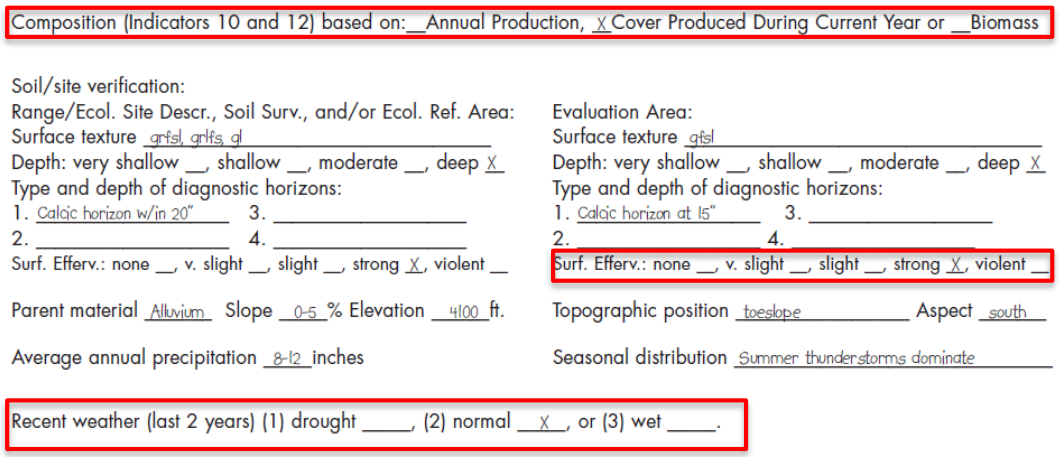
1.4. Output

choice_kind_id … smallint; 1, 2, 3, 4, and 5; Choice Kind Identifier
choice_kind … character varying(20); one per choice_kind_id; Choice Kind
Seq … smallint; Choice Sequence
Choice ID … smallint; Choice Identifier
Choice Data Text Entry … character varying(50); Choice Data Text Entry
Choice Label … character varying(50); Choice Label
Choice Description … character varying; Choice Description

Service GRAS-16i: Get Rangeland Health Assessment GRAS Choice Lists (GetRHGRASChoiceLists)

Purpose: Get and return a payload of choice lists from GRAS data mart domain tables to be utilized in populating composition basis, surface effervescence, and recent weather conditions.

Excerpt below from Interpreting Indicators of Rangeland Health, Version 4 technical reference showing data to be collected.



Composition (Indicators 10 and 12) based on: __Annual Production, X Cover Produced During Current Year or __Biomass

Soil/site verification:

Range/Ecol. Site Descr., Soil Surv., and/or Ecol. Ref. Area:	Evaluation Area:
Surface texture grfsl, grlfs, gl	Surface texture gfsl
Depth: very shallow __, shallow __, moderate __, deep X	Depth: very shallow __, shallow __, moderate __, deep X
Type and depth of diagnostic horizons:	Type and depth of diagnostic horizons:
1. Calcic horizon w/in 20" 3. _____	1. Calcic horizon at 15" 3. _____
2. _____ 4. _____	2. _____ 4. _____
Surf. Efferv.: none __, v. slight __, slight __, strong X, violent __	Surf. Efferv.: none __, v. slight __, slight __, strong X, violent __
Parent material Alluvium Slope 0-5 % Elevation 4100 ft.	Topographic position toeslope Aspect south
Average annual precipitation 8-12 inches	Seasonal distribution Summer thunderstorms dominate

Recent weather (last 2 years) (1) drought _____, (2) normal __X__, or (3) wet _____.

Service Signature

Request Payload

No data is passed into the service for processing other than requesting the service to run

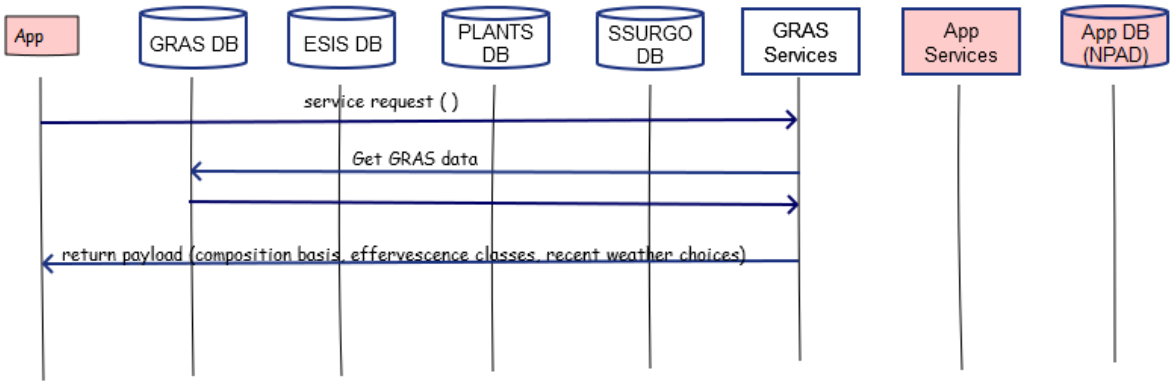
Result Payload

choice_kind_id … smallint; Choice Kind Identifier
choice_kind … character varying(20); one per choice_kind_id; Choice Kind
choice_sequence … smallint; Choice Sequence
choice_id … smallint; Choice Identifier
choice_data_entry … character varying(40); Choice Data Text Entry
choice_label … character varying(40); Choice Label
choice_description … character varying(255); Choice Description

Reference Data Sources

GRAS data mart tables
- d_rhi_composition_basis table
- d_rhi_effervescence_classes table
- d_rhi_recent_weather table

GRAS-16i: Get RH GRAS Choice Lists



Component

1. **Get Rangeland Health GRAS Choice Lists (GetRHGRAShoiceLists)**
   1.1. Inputs
   None

   1.2. Data
   GRAS data mart tables
   - d_rhi_composition_basis
     - choice_id
     - choice_sequence
     - choice_data_entry
     - choice_label
     - choice_description
     - choice_obsolete
   - d_rhi_effervescence_classes
     - choice_id
     - choice_sequence
     - choice_data_entry
     - choice_label
     - choice_description
     - choice_obsolete
   - d_rhi_recent_weather
     - choice_id
     - choice_sequence
     - choice_data_entry
     - choice_label
     - choice_description
     - choice_obsolete

   1.3. Methods
   #Get composition basis selections from GRAS d_rhi_composition_basis table to

populate choice list when basis for rating Indicators 10 and 12.

Select
choice_id
choice_sequence
choice_data_entry
choice_label
choice_description
choice_obsolete
Into rh_gras_choice_list table
From d_rhi_composition_basis table
Where choice_obsolete=False

Alter Table rh_gras_choice_list
Add compbasis as choice_kind
Add 1 as choice_kind_id

Resulting rh_gras_choice_list table
choice_kind_id=1
choice_kind=compbasis
choice_id
choice_sequence
choice_data_entry
choice_label
choice_description

 # Get effervescence classes from GRAS d_rhi_effervescence_classes table to populate choice list when selecting an effervescence class for the soil surface.
Select
choice_id
choice_sequence
choice_data_entry
choice_label
choice_description
Into rh_gras_choice_list table
From d_rhi_effervescence_classes table
Where choice_obsolete=False

Alter Table rh_gras_choice_list
Add effclass as choice_kind
Add 2 as choice_kind_id

Resulting rh_gras_choice_list table
choice_kind_id=1 and 2
choice_kind=compbasis and effclass
choice_id
choice_sequence
choice_data_entry

choice_label
choice_description

#Get recent weather condition choices from GRAS d_rhi_recent_weather table to populate choice list when weather conditions for the last two years.

Select
choice_id
choice_sequence
choice_data_entry
choice_label
choice_description
Into rh_gras_choice_list table
From d_rhi_recent_weather table
Where choice_obsolete=False

Alter Table rh_gras_choice_list
Add weacond as choice_kind
Add 3 as choice_kind_id

Resulting rh_gras_choice_list table
choice_kind_id=1, 2, and 3
choice_kind= compbasis, effclass, and weacond
choice_id
choice_sequence
choice_data_entry
choice_label
choice_description

#Send to output
Output data in rh_gras_choice_list table

#The data in rh_gras_choice_list should enable the creation of the following table.

choice_kind_id	choice_kind	choice_sequence	choice_id	choice_data_entry	choice_label	choice_description
1	compbasis	1	1	annual production	Annual Production	Indicators 10 and 12 are rated based on aboveground annual production produced during current year.
1	compbasis	2	2	cover	Cover Produced During Current Year	Indicators 10 and 12 are rated based on plant cover produced during current year.
1	compbasis	3	3	biomass	Biomass	Indicators 10 and 12 are rated based on total biomass at the site.
2	effclass	1	1	none	Noneffervescent	Few bubbles seen.
2	effclass	2	2	very slight	Very slightly effervescent	Few bubbles seen.
2	effclass	3	3	slight	Slightly effervescent	Bubbles readily seen.
2	effclass	4	4	strong	Strongly effervescent	Bubbles form low foam.

2	effclass	5	5	violent	Violently effervescent	Thick foam forms quickly.
3	weacond	1	1	drought	Drought	Below normal precipitation for the last two years.
3	weacond	2	2	normal	Normal	Normal precipitation for the last two years.
3	weacond	3	3	wet	Wet	Above average precipitation for the last two years.

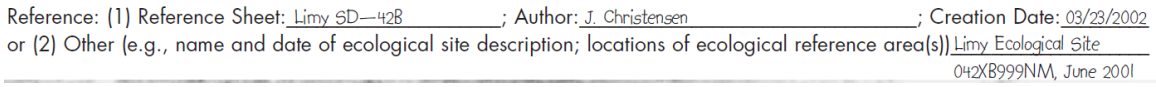
1.4. Output

choice_kind_id … smallint; Choice Kind Identifier
choice_kind … character varying(20); one per choice_kind_id; Choice Kind
choice_sequence … smallint; Choice Sequence
choice_id … smallint; Choice Identifier
choice_data_entry … character varying(40); Choice Data Text Entry
choice_label … character varying(40); Choice Label
choice_description … character varying(255); Choice Description

Service GRAS-16j: Get Rangeland Health Assessment Reference Used Information (GetRHRefUsed)

Purpose: Get and return information regarding the reference used (latest approved version of reference sheet or ecological site description) to conduct the rangeland health assessment. User will select which reference document was utilized.

Excerpt below from Interpreting Indicators of Rangeland Health, Version 4 technical reference showing the information displayed for the selected reference used.


Reference: (1) Reference Sheet: Limy SD—42B ; Author: J. Christensen ; Creation Date: 03/23/2002
or (2) Other (e.g., name and date of ecological site description; locations of ecological reference area(s)) Limy Ecological Site
042XB999NM, June 2001

Service Signature

Request Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
site_id … integer, one in the request payload, Sampling Site Identifier
rh_ref_used … integer; 1 = Reference Sheet utilized, 2 = Ecological Site Description utilized as reference; Rangeland Health Reference Used
es_id … character varying(60), one or more per FIS, Ecological Site Identifier

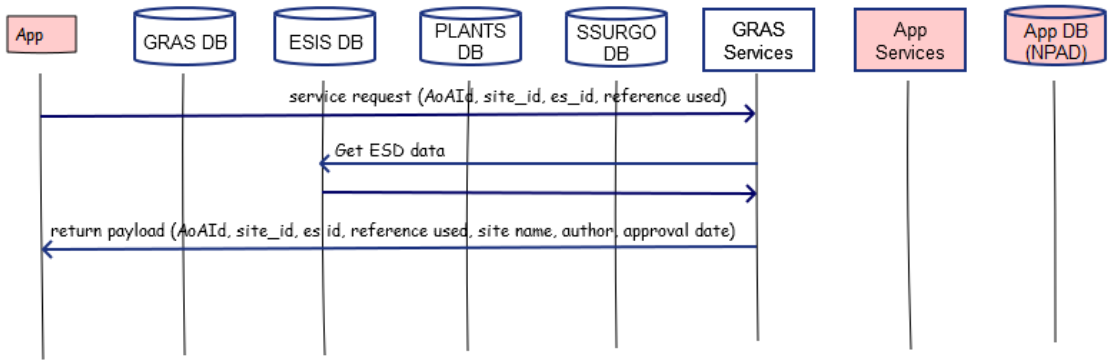
Result Payload

AoAId … integer, one in the request payload, Area of Analysis Identifier
site_id … integer, one in the request payload, Sampling Site Identifier
es_id … character varying(60), one or more per FIS, Ecological Site Identifier
rh_ref_used … integer; 1 = Reference Sheet utilized, 2 = Ecological Site Description utilized as reference; Rangeland Health Reference Used
es_range_name … character varying(120), Ecological Site Name
author … character varying(255); Author's Name
author_date … datetime; Author's Date
approval … character varying(255); Approver's Name
approval_date … datetime; Approval Date

Reference Data Sources

ESIS data mart tables
ecological_sites
site_approvals
reference_sheet

GRAS-16j: Get RH Reference Used Information



Component

1. **Get Rangeland Health Reference Used Info (GetRHRefUsed)**

   1.1. Inputs

AoAId	site_id	rh_ref_used	es_id
1	10	1	R077AY006TX

or

AoAId	site_id	rh_ref_used	es_id
1	10	2	R077AY006TX

   1.2. Data

ESIS data mart tables
- ecological_sites
  - es_type
  - es_mlra
  - es_mlru
  - es_site_number
  - es_state
  - range_site_primary_name
  - range_site_secondary_name
  - range_site_tertiary_name
- site_approvals
  - es_type
  - es_mlra
  - es_mlru
  - es_site_number
  - es_state
  - version
  - author
  - author_date
  - approval
  - approval_date

reference_sheet
es_type
es_mlra
es_mlru
es_site_number
es_state
rs_author
rs_date

reference_sheet_approvals
es_type
es_mlra
es_mlru
es_site_number
es_state
version
revision_author
revision_author_date
approval
approval_date

NPAD tables
site
site_id
es_id

1.3. Methods
For AoAId
For site_id
For es_id
#Get ecological site name, author, author date, approver, and approval date for latest version of reference sheet (reference sheet selected as the reference used).
If rh_ref_used = 1
Select
MAX(reference_sheet_approvals.version) as latest_version
Into rh_latest_version_temp table
From reference_sheet_approvals table
Where reference_sheet_approvals.approval_date IS NOT NULL

#Original reference sheet version. No revisions have been made.
If rh_lastest_version_temp.latest_version = 1
Select
ecological_sites.concatenated(es_range_primary_name, es_secondary_name, es_tertiary_name) as es_range_name
reference_sheet.rs_author as author
reference_sheet.rs_date as author_date

reference_sheet_approvals.approval
reference_sheet_approvals.approval_date
Into rh_ref_used_temp table
From ecological_sites table
Inner Join reference_sheet
On ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=reference_sheet.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)
Inner Join reference_sheet_approvals
On reference_sheet.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=reference_sheet_approvals.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)
Where ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=es_id

#Revisions have been made to the original reference sheet version.
Elseif rh_lastest_version_temp.latest_version > 1
Select
ecological_sites.concatenated(es_range_primary_name, es_secondary_name, es_tertiary_name) as es_range_name
reference_sheet_approvals.revision_author as author
reference_sheet_approvals.revision_date as author_date
reference_sheet_approvals.approval
reference_sheet_approvals.approval_date
Into rh_ref_used_temp table
From ecological_sites table
Inner Join reference_sheet_approvals
On ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=reference_sheet_approvals.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)
Where ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=es_id AND rh_latest_version_temp.latest_version=reference_sheet_approvals.version

Resulting rh_ref_used_temp table
AoAId
site_id
es_id
rh_ref_used
es_range_name
author
author_date
approval
approval_date

#Get ecological site name, author, author date, approver and approval date for latest approved version of ecological site description (ecological site description selected as the reference used).
Elseif rh_ref_used = 2
Select
MAX(site_approvals.version) as latest_version
Into rh_latest_version_temp table
From site_approvals table
Where approval_date IS NOT NULL

Select
ecological_sites.concatenated(es_range_primary_name, es_secondary_name, es_tertiary_name) as es_range_name
site_approvals.author
site_approvals.author_date
site_approvals.approval
site_approvals.approval_date
Into rh_ref_used_temp table
From ecological_sites table
Inner Join site_approvals
On ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=site_approvals.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)
Where ecological_sites.concatenated(es_type, es_mlra, es_mlru, es_site_number, es_state)=es_id AND rh_latest_version_temp.latest_version=site_approvals.version

Resulting rh_ref_used_temp table
AoAId
site_id
es_id
rh_ref_used
es_range_name
author
author_date
approval
approval_date

#Send to output
Output data in rh_ref_used_temp table

1.4. Output
AoAId … integer, one in the request payload, Area of Analysis Identifier
site_id … integer, one in the request payload, Sampling Site Identifier
es_id … character varying(60), one or more per FIS, Ecological Site Identifier

rh_ref_used … integer; 1 = Reference Sheet utilized, 2 = Ecological Site Description utilized as reference; Rangeland Health Reference Used
es_range_name … character varying(120), Ecological Site Name
author … character varying(255); Author's Name
author_date … datetime; Author's Date
approval … character varying(255); Approver's Name
approval_date … datetime; Approval Date

Service GRAS-17: Calculate Basic Roughage Balance and Accumulated Balance (CalcBasicRoughBal)

Purpose: Calculate monthly and yearly roughage balance and accumulated roughage balance. From application inputs, compute yearly beginning roughage balance and yearly harvested, bought, sold and feed roughage amounts.

This service calculates information needed for Basic level roughage report. The only difference between the Basic and Detailed roughage reports is that under the Detailed report the amount of roughage feed is allocated by AoA, whereas in the Basic report the roughage feed is allocated at the operating unit level.

Roughage/Hay (Basic)

Year	2016
Hay on Farm (Beginning Balance)	850,000
Feeding Waste %	15

○ AU ○ AUM ○ Ton ◉ LBS.

	January	February	March	April	May	June	July	August	September	October	November	December	TOTAL
Harvested	0	0	0	0	304,000	609,800	914,800	0	0	0	0	0	1,828,600
Buy	0	0	0	0	0	0	0	137,800	0	0	0	0	137,800
Sell	0	0	0	0	0	0	0	0	0	700,000	0	0	700,000
Feed	269,600	252,400	156,000	0	0	0	0	0	0	0	261,000	269,600	1,208,600
Balance	580,400	-252,400	-156,000	0	304,000	609,800	914,800	137,800	0	-700,000	-261,000	-269,600	907,800
Accumulated Balance	580,400	328,000	172,000	172,000	476,000	1,085,800	2,000,600	2,138,400	2,138,400	1,438,400	1,177,400	907,800	907,800

Service Signature

Request Payload

roughage_supply_id … integer; one in the request; Roughage Supply Identifier
starting_hay_amt_lbs … integer; Initial Pounds of Stored Roughage
year … integer; one or more per grazing system (up to 10); Roughage Supply Year; actual year (e.g. 2015, 2016, etc.)
feeding_waste_pct … integer; Percentage Waste During Feeding
month … character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December
harvested_amt_lbs … integer; Pounds of Roughage Harvested
buy_amt_lbs … integer; Pounds of Roughage Purchased
sell_amt_lbs … integer; Pounds of Roughage Sold
feed_amt_lbs ... integer; Pounds of Roughage Fed to Grazing System Animals

Note: the data in the request payload will come from the GRAS transaction database, for NRCS part of the National Planning and Agreements Database (NPAD). The data is associated to a grazing operation corresponding to a customer case file/folder, and not

necessarily to a particular grazing system. A ranch may have more than one grazing system, and the roughage supply supports all grazing systems on the ranch.

Result Payload

roughage_supply_id … integer; one in the request; Roughage Supply Identifier
year … integer; one or more per grazing system (up to 10); Roughage Supply Year; actual year (e.g. 2015, 2016, etc.)
feeding_waste_pct … integer; Percentage Waste During Feedingtotal_harvested_amt_lbs … integer; Total Annual Pounds of Roughage Harvested
total_buy_amt_lbs … integer; Total Annual Pounds of Roughage Bought
total_sell_amt_lbs … integer; Total Annual Pounds of Roughage Sold
total_feed_amt_lbs … integer; Total Annual Pounds of Roughage Fed to Grazing System Animals

month … character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December
harvested_amt_lbs… integer; Pounds of Roughage Harvested
buy_amt_lbs … integer; Pounds of Roughage Purchased
sell_amt_lbs … integer; Pounds of Roughage Sold
feed_amt_lbs … integer; Pounds of Roughage Fed to Grazing System Animals

beginning_balance_lbs … integer; Total Pounds of Roughage Available at Beginning of Each Year
jan_hay_balance_lbs … integer; Roughage Balance (Pounds) in January
feb_hay_balance_lbs … integer; Roughage Balance (Pounds) in February
mar_hay_balance_lbs … integer; Roughage Balance (Pounds) in March
apr_hay_balance_lbs … integer; Roughage Balance (Pounds) in April
may_hay_balance_lbs … integer; Roughage Balance (Pounds) in May
jun_hay_balance_lbs … integer; Roughage Balance (Pounds) in June
jul_hay_balance_lbs … integer; Roughage Balance (Pounds) in July
aug_hay_balance_lbs … integer; Roughage Balance (Pounds) in August
sep_hay_balance_lbs … integer; Roughage Balance (Pounds) in September
oct_hay_balance_lbs … integer; Roughage Balance (Pounds) in October
nov_hay_balance_lbs … integer; Roughage Balance (Pounds) in November
dec_hay_balance_lbs … integer; Roughage Balance (Pounds) in December
total_hay_balance_lbs … integer; Total Roughage Balance (Pounds) for Year

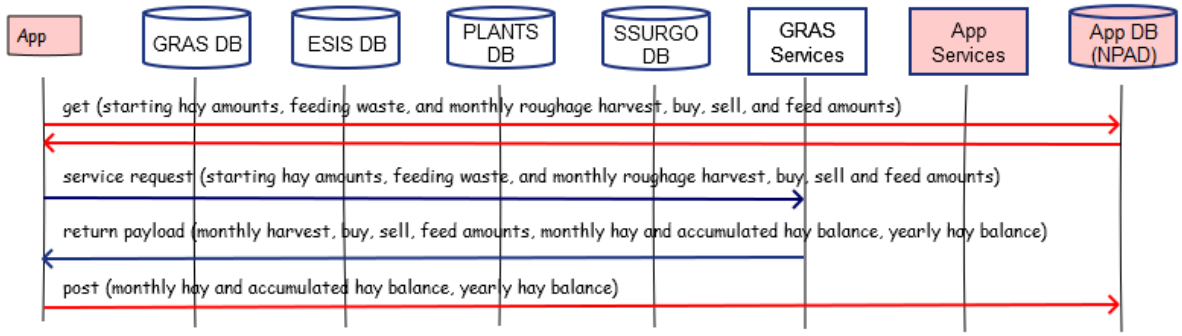
jan_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in January
feb_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in February
mar_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in March
apr_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in April

may_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in May
jun_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in June
jul_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in July
aug_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in August
sep_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in September
oct_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in October
nov_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in November
dec_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in December
total_accum_balance_lbs … integer; Total Accumulated Roughage Balance (Pounds) for Year

Reference Data Sources

NPAD database
roughage_supply table
roughage_supply_monthly table

GRAS-17: Calculate Basic Roughage Balance and Accumulated Balance



Component

1. **Calculate Basic Roughage Balance and Accumulated Balance (CalcBasicRoughBal)**
   1.1. Inputs
   roughage_supply_id … one in the request from NPAD
   starting_hay_amt_lbs … pounds, from NPAD

year … one or more from NPAD
feeding_waste_pct … percent, from NPADmonth … from NPAD
harvested_amt_lbs … pounds, from NPAD
buy_amt_lbs … pounds, from NPAD
sell_amt_lbs … pounds, from NPAD
feed_amt_lbs ... pounds, from NPAD

1.2. Methods

#Calculate yearly total harvested, bought, sold, and feed amounts
roughage_supply_id … one in the request from NPAD
For each year
For each month in the year
total_harvested_amt_lbs = total_harvested_amt_lbs + harvested_amt_lbs
total_buy_amt_lbs = total_buy_amt_lbs + buy_amt_lbs
total_sell_amt_lbs = total_sell_amt_lbs + sell_amt_lbs
total_feed_amt_lbs = total_feed_amt_lbs + feed_amt_lbs

#Pass monthly harvest, buy, sell, and feed amounts for this year from Input to Output
For each year
Output month, harvested_amt_lbs, buy_amt_lbs, sell_amt_lbs, feed_amt_lbs

#Calculate beginning roughage balance for the initial year
For initial year
beginning_balance_lbs = starting_hay_amt_lbs

#Calculate monthly hay balance and monthly accumulated hay balance
For each year
If month == January
jan_hay_balance_lbs = beginning_balance_lbs + harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - feed_amt_lbs
jan_accum_balance_lbs = jan_hay_balance_lbs
If month == February
feb_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - feed_amt_lbs
feb_accum_balance_lbs = jan_accum_balance_lbs + feb_hay_balance_lbs
If month == March
mar_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - feed_amt_lbs
mar_accum_balance_lbs = feb_accum_balance_lbs + mar_hay_balance_lbs
If month == April
apr_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - feed_amt_lbs
apr_accum_balance_lbs = mar_accum_balance_lbs + apr_hay_balance_lbs
If month == May

```
    may_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    may_accum_balance_lbs = apr_accum_balance_lbs +
    may_hay_balance_lbs
If month == June
    jun_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    jun_accum_balance_lbs = may_accum_balance_lbs + jun_hay_balance_lbs
If month == July
    jul_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    jul_accum_balance_lbs = jun_accum_balance_lbs + jul_hay_balance_lbs
If month == August
    aug_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    aug_accum_balance_lbs = jul_accum_balance_lbs + aug_hay_balance_lbs
If month == September
    sep_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    sep_accum_balance_lbs = aug_accum_balance_lbs + sep_hay_balance_lbs
If month == October
    oct_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    oct_accum_balance_lbs = sep_accum_balance_lbs + oct_hay_balance_lbs
If month == November
    nov_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    nov_accum_balance_lbs = oct_accum_balance_lbs + nov_hay_balance_lbs
If month == December
    dec_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    feed_amt_lbs
    dec_accum_balance_lbs = nov_accum_balance_lbs + dec_hay_balance_lbs
    total_hay_balance_lbs = dec_accum_balance_lbs
    total_accum_balance_lbs = dec_accum_balance_lbs

#Send monthly and total hay balance for this year to Output
For this year
     output year, roughage_supply_id, jan_hay_balance_lbs,
     feb_hay_balance_lbs, mar_hay_balance_lbs, apr_hay_balance_lbs,
     may_hay_balance_lbs, jun_hay_balance_lbs, jul_hay_balance_lbs,
     aug_hay_balance_lbs, sep_hay_balance_lbs, oct_hay_balance_lbs,
     nov_hay_balance_lbs, dec_hay_balance_lbs, total_hay_balance_lbs

#Send monthly and total accumulated hay balance for this year to Output
For this year
     output year, jan_accum_balance_lbs, feb_accum_balance_lbs,
     mar_accum_balance_lbs, apr_accum_balance_lbs,
```

may_accum_balance_lbs, jun_accum_balance_lbs,
jul_accum_balance_lbs, aug_accum_balance_lbs,
sep_accum_balance_lbs, oct_accum_balance_lbs,
nov_accum_balance_lbs, dec_accum_balance_lbs,
total_accum_hay_balance

#Calculate beginning roughage balance for next year
If total_accum_balance_lbs > 0
    beginning_balance_lbs = total_accum_balance_lbs
Else if total_accum_balance_lbs <= 0
    beginning_balanceamt_lbs = 0

#Pass feeding waste percent for this year from Input to Output
Output feeding_waste_pct

1.3. Output

roughage_supply_id … integer; one in the request; Roughage Supply Identifier
year … integer; one or more per grazing system (up to 10); Roughage Supply Year; actual year (e.g. 2015, 2016, etc.)
feeding_waste_pct … integer; Percentage Waste During Feeding
total_harvested_amt_lbs … integer; Total Annual Pounds of Roughage Harvested
total_buy_amt_lbs … integer; Total Annual Pounds of Roughage Bought
total_sell_amt_lbs … integer; Total Annual Pounds of Roughage Sold
total_feed_amt_lbs … integer; Total Annual Pounds of Roughage Fed
month … character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December
harvested_amt_lbs … integer; Pounds of Roughage Harvested
buy_amt_lbs … integer; Pounds of Roughage Purchased
sell_amt_lbs … integer; Pounds of Roughage Sold
feed_amt_lbs … integer; Pounds of Roughage Fed to Grazing System Animals

beginning_balance_lbs … integer; Total Pounds of Roughage Available at Beginning of Each Year
jan_hay_balance_lbs … integer; Roughage Balance (Pounds) in January
feb_hay_balance_lbs … integer; Roughage Balance (Pounds) in February
mar_hay_balance_lbs … integer; Roughage Balance (Pounds) in March
apr_hay_balance_lbs … integer; Roughage Balance (Pounds) in April
may_hay_balance_lbs … integer; Roughage Balance (Pounds) in May
jun_hay_balance_lbs … integer; Roughage Balance (Pounds) in June
jul_hay_balance_lbs … integer; Roughage Balance (Pounds) in July
aug_hay_balance_lbs … integer; Roughage Balance (Pounds) in August
sep_hay_balance_lbs … integer; Roughage Balance (Pounds) in September
oct_hay_balance_lbs … integer; Roughage Balance (Pounds) in October
nov_hay_balance_lbs … integer; Roughage Balance (Pounds) in November
dec_hay_balance_lbs … integer; Roughage Balance (Pounds) in December
total_hay_balance_lbs … integer; Total Roughage Balance (Pounds) for Year

jan_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in January
feb_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in February
mar_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in March
apr_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in April
may_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in May
jun_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in June
jul_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in July
aug_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in August
sep_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in September
oct_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in October
nov_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in November
dec_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in December
total_accum_balance_lbs … integer; Total Accumulated Roughage Balance (Pounds) for Year

Service GRAS-18: Calculate Detail Roughage Balance and Accumulated Balance (CalcBasicRoughBal)

Purpose: Calculate monthly and yearly roughage balance and accumulated roughage balance. From application inputs, compute yearly beginning roughage balance and yearly harvested, bought, sold and feed roughage amounts.

This service calculates information needed for Detail level roughage report. The only difference between the Basic and Detailed roughage reports is that under the Detailed report the amount of roughage feed is allocated by AoA, whereas in the Basic report the roughage feed is allocated at the operating unit level.

Roughage/Hay (Detailed)

AU AUM Ton LBS.

Year	2016
Hay on Farm (Beginning Balance)	850,000
Feeding Waste %	15

PLU	Acres		January	February	March	April	May	June	July	August	September	October	November	December	TOTAL
		Harvested	0	0	0	0	304,000	609,800	914,800	0	0	0	0	0	1,828,600
		Buy	0	0	0	0	0	0	0	137,800	0	0	0	0	137,800
		Sell	0	0	0	0	0	0	0	0	0	700,000	0	0	700,000
1	200	Feed	105,000	100,400	63,000	0	0	0	0	0	0	0	101,000	105,000	474,400
2	120	Feed	72,000	67,000	38,000	0	0	0	0	0	0	0	70,000	72,000	319,000
3	160	Feed	92,600	85,000	55,000	0	0	0	0	0	0	0	90,000	92,600	415,200
		Total Feed	269,600	252,400	156,000	0	0	0	0	0	0	0	261,000	269,600	1,208,600
		Balance	580,400	-252,400	-156,000	0	304,000	609,800	914,800	137,800	0	-700,000	-261,000	-269,600	907,800
		Accumulated Balance	580,400	328,000	172,000	172,000	476,000	1,085,800	2,000,600	2,138,400	2,138,400	1,438,400	1,177,400	907,800	907,800

Service Signature

Request Payload

roughage_supply_id … integer; one in the request; Roughage Supply Identifier
starting_hay_amt_lbs … integer; Initial Pounds of Stored Roughage
year … integer; one or more per grazing system (up to 10); Roughage Supply Year; actual year (e.g. 2015, 2016, etc.)
feeding_waste_pct … integer; Percentage Waste During Feeding month … character varying(30);
Month of the Grazing Year; e.g. January, February, etc. through December
harvested_amt_lbs … integer; Pounds of Roughage Harvested
buy_amt_lbs … integer; Pounds of Roughage Purchased
sell_amt_lbs … integer; Pounds of Roughage Sold
AoAID … integer, one or more in the request payload, Area of Analysis Identifier
month … character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December

feed_amt_lbs ... integer; Pounds of Roughage Fed to Grazing System Animals

Note: the data in the request payload will come from the GRAS transaction database, for NRCS part of the National Planning and Agreements Database (NPAD). The data is associated to a grazing operation corresponding to a customer case file/folder, and not necessarily to a particular grazing system. A ranch may have more than one grazing system, and the roughage supply supports all grazing systems on the ranch.

Result Payload

roughage_supply_id … integer; one in the request; Roughage Supply Identifier
year … integer; one or more per grazing system (up to 10); Roughage Supply Year; actual year (e.g. 2015, 2016, etc.)
feeding_waste_pct … integer; Percentage Waste During Feeding
total_harvested_amt_lbs … integer; Total Annual Pounds of Roughage Harvested
total_buy_amt_lbs … integer; Total Annual Pounds of Roughage Bought
total_sell_amt_lbs … integer; Total Annual Pounds of Roughage Sold

month … character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December
harvested_amt_lbs … integer; Pounds of Roughage Harvested
buy_amt_lbs … integer; Pounds of Roughage Purchased
sell_amt_lbs … integer; Pounds of Roughage Sold

AoAID integer, one or more in the request payload, Area of Analysis Identifier
total_feed_amt_lbs … integer; Total Annual Pounds of Roughage Fed in the AoA

month … character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December
feed_amt_lbs … integer; Pounds of Roughage Fed to Grazing System Animals

jan_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in January
feb_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in February
mar_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in March
apr_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in April
may_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in May
jun_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in June
jul_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in July
aug_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in August
sep_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in September
oct_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in October
nov_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in November
dec_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in December
this_year_total_feed_amt_lbs … integer; Total Annual Pounds of Roughage Fed

beginning_balance_lbs … integer; Total Pounds of Roughage Available at Beginning of Each Year
jan_hay_balance_lbs … integer; Roughage Balance (Pounds) in January
feb_hay_balance_lbs … integer; Roughage Balance (Pounds) in February
mar_hay_balance_lbs … integer; Roughage Balance (Pounds) in March
apr_hay_balance_lbs … integer; Roughage Balance (Pounds) in April
may_hay_balance_lbs … integer; Roughage Balance (Pounds) in May
jun_hay_balance_lbs … integer; Roughage Balance (Pounds) in June
jul_hay_balance_lbs … integer; Roughage Balance (Pounds) in July
aug_hay_balance_lbs … integer; Roughage Balance (Pounds) in August
sep_hay_balance_lbs … integer; Roughage Balance (Pounds) in September
oct_hay_balance_lbs … integer; Roughage Balance (Pounds) in October
nov_hay_balance_lbs … integer; Roughage Balance (Pounds) in November
dec_hay_balance_lbs … integer; Roughage Balance (Pounds) in December
total_hay_balance_lbs … integer; Total Roughage Balance (Pounds) for Year

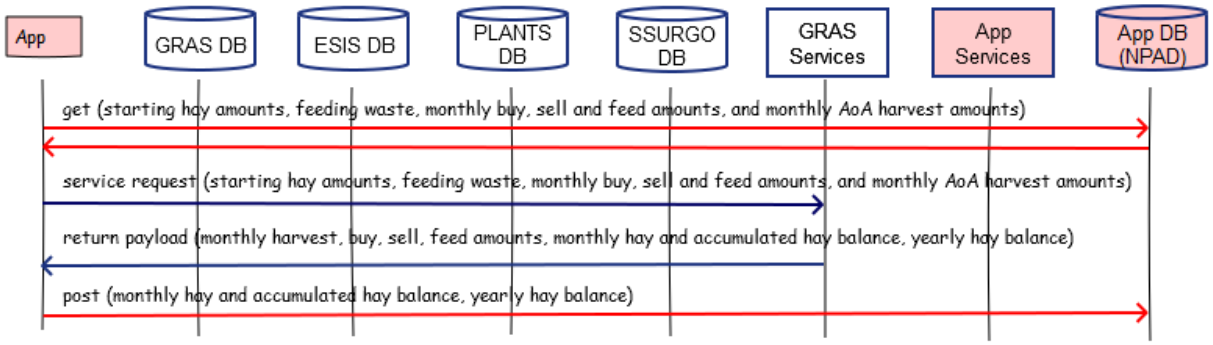
jan_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in January
feb_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in February
mar_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in March
apr_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in April
may_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in May
jun_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in June
jul_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in July
aug_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in August
sep_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in September
oct_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in October
nov_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in November
dec_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in December
total_accum_balance_lbs … integer; Total Accumulated Roughage Balance (Pounds) for Year

Reference Data Sources

NPAD database

roughage_supply table
roughage_supply_monthly table
roughage_supply_monthly_detail table

GRAS-18: Calculate Detailed Roughage Balance and Accumulated Balance



Component

1. Calculate Detail Roughage Balance and Accumulated Balance (CalcDetailRoughBal)

1.1. Inputs

roughage_supply_id … one in the request from NPAD
starting_hay_amt_lbs … pounds, from NPAD
year … one or more from NPAD
feeding_waste_pct … percent, from NPAD
month … from NPAD
harvested_amt_lbs … pounds, from NPAD
buy_amt_lbs … pounds, from NPAD
sell_amt_lbs … pounds, from NPAD
AoA Identifier … one or more
month
feed_amt_lbs

1.2. Methods

#Calculate yearly total harvested, bought, and sold amounts
For roughage_supply_id
For each year
For each month in the year
total_harvested_amt_lbs = total_harvested_amt_lbs + harvested_amt_lbs
total_buy_amt_lbs = total_buy_amt_lbs + buy_amt_lbs
total_sell_amt_lbs = total_sell_amt_lbs + sell_amt_lbs

#Calculate yearly total fed amount by AoA
For each year
For each AoA
For each month in the year
total_feed_amt_lbs = total_feed_amt_lbs + feed_amt_lbs

```
#Calculate total monthly fed amounts for all AoAs for each year
For each year
  For each AoA
      If month == January
          jan_total_feed_amt_lbs = jan_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + jan_total_feed_amt_lbs
      If month == February
          feb_total_feed_amt_lbs = feb_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + feb_total_feed_amt_lbs
      If month == March
          mar_total_feed_amt_lbs = mar_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + mar_total_feed_amt_lbs
      If month == April
          apr_total_feed_amt_lbs = apr_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + apr_total_feed_amt_lbs
      If month == May
          may_total_feed_amt_lbs = may_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + may_total_feed_amt_lbs
      If month == June
          jun_total_feed_amt_lbs = jun_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + jun_total_feed_amt_lbs
      If month == July
          jul_total_feed_amt_lbs = jul_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + jul_total_feed_amt_lbs
      If month == August
          aug_total_feed_amt_lbs = aug_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + aug_total_feed_amt_lbs
      If month == September
          sep_total_feed_amt_lbs = sep_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + sep_total_feed_amt_lbs
      If month == October
          oct_total_feed_amt_lbs = oct_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + oct_total_feed_amt_lbs
      If month == November
          nov_total_feed_amt_lbs = nov_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + nov_total_feed_amt_lbs
      If month == December
          dec_total_feed_amt_lbs = dec_total_feed_amt_lbs + feed_amt_lbs
          cumulative_feed_amt = cumulative_feed_amt + dec_total_feed_amt_lbs

      #Total roughage feed this year
      this_year_total_feed_amt_lbs = cumulative_feed_amt

          #Reset cumulative feed amount to zero for next year
          cumulative_feed_amt = 0.00

      #Send monthly total feed amounts and total yearly feed amounts for this
```

year to Output
For this year
output year, roughage_supply_id, jan_total_feed_amt_lbs, feb_total_feed_amt_lbs, mar_total_feed_amt_lbs, apr_total_feed_amt_lbs, may_total_feed_amt_lbs, jun_total_feed_amt_lbs, jul_total_feed_amt_lbs, aug_total_feed_amt_lbs, sep_ total_feed_amt_lbs, oct_total_feed_amt_lbs, nov_total_feed_amt_lbs, dec_total_feed_amt_lbs, this_year_total_feed_amt_lbs

#Pass monthly harvested amounts for AoA for this year from Input to Output
Output month, harvested_amt_lbs

#Calculate beginning roughage balance for initial year
For initial year
beginning_balance_lbs = starting_hay_amt_lbs

#Calculate monthly hay balance and monthly accumulated hay balance
For each year
If month == January
jan_hay_balance_lbs = beginning_balance_lbs + harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs – jan_total_feed_amt_lbs
jan_accum_balance_lbs = jan_hay_balance_lbs
If month == February
feb_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - feb_total_feed_amt_lbs
feb_accum_balance_lbs = jan_accum_balance_lbs + feb_hay_balance_lbs
If month == March
mar_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - mar_total_feed_amt_lbs
mar_accum_balance_lbs = feb_accum_balance_lbs + mar_hay_balance_lbs
If month == April
apr_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - apr_total_feed_amt_lbs
apr_accum_balance_lbs = mar_accum_balance_lbs + apr_hay_balance_lbs
If month == May
may_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - may_total_feed_amt_lbs
may_accum_balance_lbs = apr_accum_balance_lbs + may_hay_balance_lbs
If month == June
jun_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs - jun_total_feed_amt_lbs
jun_accum_balance_lbs = may_accum_balance_lbs + jun_hay_balance_lbs
If month == July

```
    jul_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    jul_total_feed_amt_lbs
    jul_accum_balance_lbs = jun_accum_balance_lbs + jul_hay_balance_lbs
If month == August
    aug_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    aug_total_feed_amt_lbs
    aug_accum_balance_lbs = jul_accum_balance_lbs + aug_hay_balance_lbs
If month == September
    sep_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    sep_total_feed_amt_lbs
    sep_accum_balance_lbs = aug_accum_balance_lbs + sep_hay_balance_lbs
If month == October
    oct_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    oct_total_feed_amt_lbs
    oct_accum_balance_lbs = sep_accum_balance_lbs + oct_hay_balance_lbs
If month == November
    nov_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    nov_total_feed_amt_lbs
    nov_accum_balance_lbs = oct_accum_balance_lbs + nov_hay_balance_lbs
If month == December
    dec_hay_balance_lbs = harvested_amt_lbs + buy_amt_lbs - sell_amt_lbs -
    dec_total_feed_amt_lbs
    dec_accum_balance_lbs = nov_accum_balance_lbs + dec_hay_balance_lbs
    total_hay_balance_lbs = dec_accum_balance_lbs
    total_accum_balance_lbs = dec_accum_balance_lbs

#Send monthly and total hay balance for this year to Output
For this year
     output year, jan_hay_balance_lbs, feb_hay_balance_lbs,
     mar_hay_balance_lbs, apr_hay_balance_lbs, may_hay_balance_lbs,
     jun_hay_balance_lbs, jul_hay_balance_lbs, aug_hay_balance_lbs,
     sep_hay_balance_lbs, oct_hay_balance_lbs, nov_hay_balance_lbs,
     dec_hay_balance_lbs, total_hay_balance_lbs

#Send monthly and total accumulated hay balance for this year to Output
For this year
     output year, jan_accum_balance_lbs, feb_accum_balance_lbs,
     mar_accum_balance_lbs, apr_accum_balance_lbs,
     may_accum_balance_lbs, jun_accum_balance_lbs,
     jul_accum_balance_lbs, aug_accum_balance_lbs,
     sep_accum_balance_lbs, oct_accum_balance_lbs,
     nov_accum_balance_lbs, dec_accum_balance_lbs,
     total_accum_hay_balance

    #Calculate beginning roughage balance for next year
    If total_accum_balance_lbs > 0
        beginning_roughage_lbs = total_accum_balance_lbs
```

Else if total_accum_balance_lbs <= 0
beginning_roughage_lbs = 0

#Pass feeding waste percent for this year from Input to Output
Output feeding_waste_pct

#Pass monthly buy, sell, and feed amounts for this year from Input to Output
Output month, buy_amt_lbs, sell_amt_lbs, feed_amt_lbs

1.3. Output

roughage_supply_id … integer; one in the request; Roughage Supply Identifier
year … integer; one or more per grazing system (up to 10); Roughage Supply Year; actual year (e.g. 2015, 2016, etc.)
feeding_waste_pct … integer; Percentage Waste During Feeding
total_harvested_amt_lbs … integer; Total Annual Pounds of Roughage Harvested
total_buy_amt_lbs … integer; Total Annual Pounds of Roughage Bought
total_sell_amt_lbs … integer; Total Annual Pounds of Roughage Sold

month … character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December
harvested_amt_lbs … integer; Pounds of Roughage Harvested
buy_amt_lbs … integer; Pounds of Roughage Purchased
sell_amt_lbs … integer; Pounds of Roughage Sold

AoAId … integer, one or more in the request payload, Area of Analysis Identifier
total_feed_amt_lbs … integer; Total Annual Pounds of Roughage Fed in the AoA

month… character varying(30); Month of the Grazing Year; e.g. January, February, etc. through December
feed_amt_lbs … integer; Pounds of Roughage Fed to Grazing System Animals

jan_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in January
feb_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in February
mar_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in March
apr_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in April
may_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in May
jun_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in June
jul_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in July
aug_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in August
sep_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in September
oct_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in October
nov_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in November
dec_total_feed_amt_lbs … integer; Total Pounds of Roughage Fed in December
this_year_total_feed_amt_lbs … integer; Total Annual Pounds of Roughage Fed

beginning_balance_lbs … integer; Total Pounds of Roughage Available at Beginning of Each Year

jan_hay_balance_lbs … integer; Roughage Balance (Pounds) in January
feb_hay_balance_lbs … integer; Roughage Balance (Pounds) in February
mar_hay_balance_lbs … integer; Roughage Balance (Pounds) in March
apr_hay_balance_lbs … integer; Roughage Balance (Pounds) in April
may_hay_balance_lbs … integer; Roughage Balance (Pounds) in May
jun_hay_balance_lbs … integer; Roughage Balance (Pounds) in June
jul_hay_balance_lbs … integer; Roughage Balance (Pounds) in July
aug_hay_balance_lbs … integer; Roughage Balance (Pounds) in August
sep_hay_balance_lbs … integer; Roughage Balance (Pounds) in September
oct_hay_balance_lbs … integer; Roughage Balance (Pounds) in October
nov_hay_balance_lbs … integer; Roughage Balance (Pounds) in November
dec_hay_balance_lbs … integer; Roughage Balance (Pounds) in December
total_hay_balance_lbs … integer; Total Roughage Balance (Pounds) for Year

jan_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in January
feb_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in February
mar_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in March
apr_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in April
may_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in May
jun_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in June
jul_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in July
aug_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in August
sep_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in September
oct_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in October
nov_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in November
dec_accum_balance_lbs … integer; Accumulated Roughage Balance (Pounds) in December
total_accum_balance_lbs … integer; Total Accumulated Roughage Balance (Pounds) for Year

Appendix I – Grassland Resource Analysis System (GRAS) Data Model

The data model at the end of this appendix was provided by NRCS and goes back to the model used to build the GRAS application in the prototype Conservation Desktop in 2012-2013. The model needs adjustment to fix gap and errors and accommodate requirements not addressed in the prior effort. However, for this version of the service specifications it has provided the primary data model reference.

The GRAS data model contains both domain and transaction tables. The domain tables, with a few exceptions become the GRAS natural resource data mart containing animal, forage, and other grazing related resource data maintained by data stewards in much the same way as soil (SSURGO), ecological site (ESIS), and other natural resource data marts. The GRAS transaction tables become part of the NRCS National Planning and Agreements Database (NPAD).

Following are the tables and their data element in GRAS data mart

d_animal_unit table

Generated: 1/4/2106 9:01:49 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
animal_unit_id	integer	Yes	Yes
animal_kind	character varying(50)	Yes	No
animal_class	character varying(50)	Yes	No
animal_gender	character varying(50)	Yes	No
animal_growth_category	character varying(50)	Yes	No
animal_default_AUE	numeric	Yes	No
animal_avg_dailyIntake_pct_bodyWeight	numeric	Yes	No
end_date	date	No	No
start_date	date	No	No
last_change_date	date	No	No
animal_default_gestation_period_days	integer	No	No

Constraints

Name	Type	Definition
d_animal_unit_pkey	Primary key	(animal_unit_id)

d_faa_water_adjustment_factor

Generated: 1/4/2106 9:02:39 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
adj_factor_id	smallint	Yes	Yes
min_adj_extent	integer	Yes	No
max_adj_extent	integer	Yes	No
adj_factor	numeric	Yes	No

Constraints

Name	Type	Definition
d_faa_water_adjustment_factor_pkey	Primary key	(adj_factor_id)

d_forage_adjustment_category table

Generated: 1/4/2106 9:03:13 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
forage_adjustment_category_id	integer	Yes	Yes
forage_adjustment_category_name	character varying(50)	Yes	No

Constraints

Name	Type	Definition
d_forage_adjustment_category_pkey	Primary key	(forage_adjustment_category_id)

d_forage_partition_activity_type table

Generated: 1/4/2106 9:03:45 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
forage_partition_activity_type_id	integer	Yes	Yes
forage_partition_activity_type_name	character varying(50)	Yes	No
forage_partition_activity_type_description	character varying(255)	No	No

Constraints

Name	Type	Definition
d_forage_partition_activity_pkey	Primary key	(forage_partition_activity_type_id)

d_forage_partition_activity_type_national_defaults table

Generated: 1/4/2106 9:04:14 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
forage_partition_activity_default_harvest_efficiency_id	integer	Yes	Yes
forage_partition_activity_type_id	integer	No	No
land_use_id	integer	No	No
land_use_name	character varying(50)	No	No
default_harvest_efficiency_pct	integer	No	No
min_allowed_efficiency_pct	integer	No	No
max_allowed_efficiency_pct	integer	No	No

Constraints

Name	Type	Definition
d_default_harvest_efficiency_pkey	Primary key	(forage_partition_activity_default_harvest_efficiency_id)
d_default_harvest_efficiency_forage_partition_activity_id_fkey	Foreign key	(forage_partition_activity_type_id) REFERENCES d_forage_partition_activity_type (forage_partition_activity_type_id)

d_forage_partition_profile table

Generated: 1/4/2106 9:05:22 PM

Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
forage_partition_profile_id	uuid	Yes	Yes
state_county_code	character(5)	No	No
plu_default_activity_ind	character(1)	No	No
forage_partition_profile_name	character varying(100)	Yes	No
forage_partition_profile_description	character varying(1000)	No	No
start_date	date	No	No
end_date	date	No	No

Constraints

Name	Type	Definition
d_forage_partition_profile_pkey	Primary key	(forage_partition_profile_id)

d_forage_partition_profile_activity table

Generated: 1/4/2106 9:05:58 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
forage_partition_profile_id	uuid	Yes	Yes
forage_partition_profile_activity_id	uuid	Yes	Yes
forage_partition_activity_type_id	integer	Yes	No
land_use_id	integer	No	No
harvest_efficiency_pct	integer	Yes	No
calendar_start_date	character(5)	No	No
calendar_end_day	character(5)	Yes	No

Constraints

Name	Type	Definition
d_forage_partition_profile_activity_pkey	Primary key	(forage_partition_profile_id, forage_partition_profile_activity_id)
d_forage_partition_profile_ac_forage_partition_activity_ty_fkey	Foreign key	(forage_partition_activity_type_id) REFERENCES d_forage_partition_activity_type

		(forage_partition_activity_type_id)
d_forage_partition_profile_act_forage_partition_profile_id_fkey	Foreign key	(forage_partition_profile_id) REFERENCES d_forage_partition_profile (forage_partition_profile_id) MATCH

d_forage_production_estimate_source table

Generated: 1/4/2106 9:06:24 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
forage_production_estimate_id	integer	Yes	Yes
forage_production_estimate_source_name	character varying(20)	Yes	No
forage_production_estimate_source_display	character varying(100)	No	No

Constraints

Name	Type	Definition
d_forage_production_estimate_source_pkey	Primary key	(forage_production_estimate_id)

d_gras_units table

Generated: 1/4/2106 9:07:15 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
gras_unit_id	integer	Yes	Yes
gras_unit_use_id	integer	No	No

gras_unit_use	character varying(30)	No	No
gras_unit_short_name	character varying(20)	No	No
gras_unit_long_name	character varying(50)	No	No
gras_unit_description	character varying(100)	No	No

Constraints

Name	Type	Definition
d_gras_units_pkey	Primary key	(gras_unit_id)

d_pci_indicators table

Generated: 1/4/2106 8:58:46 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
pci_indicators_id	smallint	Yes	Yes
pci_indicators_sequence	smallint	Yes	No
pci_indicators_name	character varying(50)	Yes	No
pci_indicators_description	character varying(255)	Yes	No
pci_indicators_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
pci_indicators primary key	Primary key	(pci_indicators_id)

d_pci_indicator_ratings table

Generated: 1/4/2106 9:07:46 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
choice_id	smallint	Yes	Yes
choice_kind_id	smallint	Yes	No
choice_kind_name	character varying(40)	Yes	No

choice_sequence	smallint	Yes	No
choice_data_value	smallint	Yes	No
choice_label	smallint	Yes	No
choice_description	character varying(255)	Yes	No
choice_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
pci indicator ratings primary key	Primary key	(choice_id)
pci indicator ratings foreign key	Foreign key	(choice_kind_id) REFERENCES d_pci_indicators (pci_indicators_id) MATCH SIMPLE ON UPDATE NO

d_plant_growth_curve_state_local table

Generated: 1/4/2106 9:08:37 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
growth_curve_id	character varying(10)	Yes	Yes
state_county_code	character(5)	Yes	No
growth_curve_name	character varying(100)	Yes	No
growth_curve_description	character varying(1000)	No	No
percent_production_jan	numeric(3)	Yes	No
percent_production_feb	numeric(3)	Yes	No
percent_production_mar	numeric(3)	Yes	No
percent_production_apr	numeric(3)	Yes	No
percent_production_may	numeric(3)	Yes	No
percent_production_jun	numeric(3)	Yes	No
percent_production_jul	numeric(3)	Yes	No
percent_production_aug	numeric(3)	Yes	No
percent_production_sep	numeric(3)	Yes	No
percent_production_oct	numeric(3)	Yes	No
percent_production_nov	numeric(3)	Yes	No
percent_production_dec	numeric(3)	Yes	No
last_changed_by	character(25)	No	No
last_change_date	timestamp without time zone	No	No

Constraints

Name	Type	Definition

d_plant_growth_curve_state_local_pkey	Primary key	(growth_curve_id)

d_rha_diag_feat_kind table

Generated: 1/4/2106 9:09:14 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
seq	integer	Yes	No
choice_id	integer	Yes	No
choice_data_entry_text	character varying(60)	Yes	No
choice_label	character varying(60)	Yes	No
choice_description	character varying(500)	No	No
obsolete	character varying(5)	Yes	No

d_rha_hillslope_profile table

Generated: 1/4/2106 9:09:45 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
seq	integer	Yes	No
choice_id	integer	Yes	No
choice_data_entry_text	character varying(60)	Yes	No
choice_label	character varying(60)	Yes	No
choice_description	character varying(500)	No	No
obsolete	character varying(6)	Yes	No

d_rha_terms_in_lieu_of_texture table

Generated: 1/4/2106 9:10:10 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
seq	integer	Yes	No
choice_id	integer	Yes	No
choice_data_entry_text	character varying(60)	Yes	No
choice_label	character varying(60)	Yes	No
choice_description	character varying(500)	No	No
obsolete	character varying(6)	Yes	No

d_rha_texture_class table

Generated: 1/4/2106 9:11:53 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
seq	integer	Yes	No
choice_id	integer	Yes	No
choice_data_entry_text	character varying(60)	Yes	No
choice_label	character varying(60)	Yes	No
choice_description	character varying(500)	No	No
obsolete	character varying(6)	Yes	No

d_rha_texture_modifier table

Generated: 1/4/2106 9:12:17 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
seq	integer	Yes	No
choice_id	integer	Yes	No
choice_data_entry_text	character varying(60)	Yes	No
choice_label	character varying(60)	Yes	No

choice_description	character varying(1000)	No	No
obsolete	character varying(6)	Yes	No

d_rhi_attribute_ratings table

Generated: 1/4/2106 9:12:49 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
choice_id	smallint	Yes	Yes
choice_sequence	smallint	Yes	No
choice_data_value	smallint	Yes	No
choice_label	character varying(25)	Yes	No
choice_short_label	character varying(5)	No	No
choice_description	character varying(255)	Yes	No
choice_obsolete	boolean	No	No

Constraints

Name	Type	Definition
d_rhi_attribute_ratings_pk	Primary key	(choice_id)

d_rhi_attributes table

Generated: 1/4/2106 9:13:23 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
rhi_attributes_id	smallint	Yes	Yes
rhi_attributes_sequence	smallint	Yes	No
rhi_attributes_name	character varying(25)	Yes	No
rhi_attributes_description	character varying(255)	Yes	No
rhi_attributes_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
rhi_attributes_pk	Primary key	(rhi_attributes_id)

d_rhi_composition_basis table

Generated: 1/4/2106 9:13:23 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
choice_id	smallint	Yes	Yes
choice_sequence	smallint	Yes	No
choice_data_entry	character varying(40)	Yes	No
choice_label	character varying(40)	Yes	No
choice_description	character varying(255)	Yes	No
choice_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
d_rhi_composition_basis_pkey	Primary key	(choice_id)

d_rhi_effervescence_classes table

Generated: 1/4/2106 9:14:20 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
choice_id	smallint	Yes	Yes
choice_sequence	smallint	Yes	No
choice_data_entry	character varying(40)	Yes	No
choice_label	character varying(40)	Yes	No
choice_description	character varying(255)	Yes	No
choice_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
d_rhi_effervescence_classes_pkey	Primary key	(choice_id)

d_rhi_ind_attr_assn table

Generated: 1/4/2106 9:14:49 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
rhi_ind_attr_assn_id	smallint	Yes	Yes
rhi_indicators_id	smallint	Yes	No
rhi_sss_assn	boolean	Yes	No
rhi_hf_assn	boolean	Yes	No
rhi_bi_assn	boolean	Yes	No
rhi_ind_attr_assn_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
d_rhi_ind_attr_assn_pk	Primary key	(rhi_ind_attr_assn_id)
d_rhi_ind_attr_assn_fk	Foreign key	(rhi_indicators_id) REFERENCES d_rhi_indicators (rhi_indicators_id) MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION

d_rhi_indicator_ratings table

Generated: 1/4/2106 9:15:16 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
choice_id	smallint	Yes	No
choice_kind_id	smallint	Yes	No
choice_kind_name	character varying(75)	Yes	No

choice_sequence	smallint	Yes	No
choice_data_value	smallint	Yes	No
choice_label	character varying(50)	Yes	No
choice_short_label	character varying(5)	Yes	No
choice_description	character varying(255)	Yes	No
choice_obsolete	boolean	Yes	No

d_rhi_indicators table

Generated: 1/4/2106 9:15:45 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
rhi_indicators_id	smallint	Yes	Yes
rhi_indicators_sequence	smallint	Yes	No
rhi_indicators_name	character varying(75)	Yes	No
rhi_indicators_description	character varying(255)	Yes	No
rhi_indicators_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
d_rhi_indicators_pk	Primary key	(rhi_indicators_id)

d_rhi_recent_weather table

Generated: 1/4/2106 9:16:07 PM
Server: PostgreSQL 9.4 (localhost:5432)
Database: Grassland Resource Analysis System (GRAS)
Schema: public

Columns

Name	Data type	Not Null?	Primary key?
choice_id	smallint	Yes	Yes
choice_sequence	smallint	Yes	No
choice_data_entry	character varying(40)	Yes	No
choice_label	character varying(40)	Yes	No

choice_description	character varying(255)	Yes	No
choice_obsolete	boolean	Yes	No

Constraints

Name	Type	Definition
d_rhi_recent_weather_pkey	Primary key	(choice_id)