The user interface enables the user to link livestock to none, one or two housing types and then choose where the manure is stored. For some housing types, only one manure type is produced and so the user has to choose from one or more types of manure storage. In other cases, two manure types are produced, in which case, the user must choose storage types for each of the manures. The user interface then outputs this information in the farm file.

In the farm file, each livestock type that is housed contains a tag Housing. Under the Housing tag, there is one or more ManureRecipient tags.

ManureRecipient

All housings have the possibility to have 0 to many Manure Recipient. Each Manure Recipient need to have unique Identity, a StorageType and a Species\_group, so the xml will look like

<ManureRecipient>

<Identity>1</Identity>

<StorageType>7</StorageType>

<Species\_group>1</ Species\_group>

</ManureRecipient>

Table 1 shows the list of manure storage types and table 2 the list of species groups

Table 1

|  |  |
| --- | --- |
| Storage types | Description |
| 1 | Slurry Tank |
| 2 | slurryTank - with cover |
| 3 | Dunghill without cover |
| 4 | Dunghill with cover |
| 5 | Biogas plant |
| 6 | Partially-separated liquid fraction, no cover |
| 7 | Partially-separated solid fraction, no cover |
| 8 |  |
| 9 | Partially-separated liquid fraction, with cover |
| 10 | Partially-separated solid fraction, with cover |
| 11 | slurryTank - with tank acidification |
| 12 | slurryTank – acidified |
| 13 | slurryTank - acidified slurry |
|  |  |

Table 2 Species groups

|  |  |
| --- | --- |
| Species Group | Description |
| 1 | Cattle |
| 2 | Pig |
| 3 | sheep |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |

Each house has its own List<GlobalVars.manurestoreRecord> manurestoreDetails. In public housing(int aHousingType, livestock aLivestock, int houseIndex, int zoneNr, string aParens), we look for to match the HousingType parameter with the HousingType tag under Housing in parameters.xml. After reading the housing parameters, we retrieve the information about the manure recipients from manurestoreRecord and create a manureStore for each ManureRecipient and added that information to the manurestoreDetails. The public manureStore(int manureStorageType, int livestockSpeciesGroup, int zoneNr, string aParens) stores the type of manure storage, as defined in Table 1 and the species group as defined in Table 2. It then loads the manure storage parameters using public void getParameters(int zoneNr).

In getParameters(int zoneNr), the model goes through the manure stores listed in parameters.xml under the ManureStorage tag, looking to match the StorageType and SpeciesGroup tags with the storageType and speciesGroup variables of the manure storage object. Having found the correct manure store, the parameters are read. This includes the manureTypeStored parameter, which is read from the TypeStored tag in parameters.xml. A list of the manureTypeStored that may be stored is shown in Table 3. Note that some types that are not relevant for livestock farms.

Table 3 Manure types

|  |  |
| --- | --- |
| Manure types | Manure type ID |
| None | 0 |
| Cattle slurry | 1 |
| Cattle/sheep deep litter | 2 |
| Grazing | 3 |
| Cattle compost | 4 |
| Solid from cattle | 5 |
| Liquid fraction | 6 |
| Pig slurry | 7 |
| Pig deep litter | 8 |
| Mink slurry | 10 |
| Solid poultry manure | 11 |
| Degassed biomasse | 12 |
| Acidified slurry and other processed manure | 13 |
| Mixed slurry | 14 |
| Sewage sludge | 15 |
| Composted household waste | 16 |
| Potatojuice | 17 |
| Press juice from green pellet production | 18 |
| Garden- and park refuse | 19 |
| Other types of organic manure | 20 |
| Solid fraction from decantercentrifuge | 21 |
| Liquid fraction from decantercentrifuge | 22 |
| Poultry slurry | 23 |
| Sheep deep litter | 24 |

Details of the manure stored in a particular storage are held in the theManure object, which is an instance of the manure class and a member of manureStore. This class stores information on whether the manure is a solid (read from the StoresSolid tag under the ManureStorage tag of parameters.xml). The model then looks in the fertMan.xml file to find the relevant manure. It does so by matching the SpeciesGroup and ManureType tags with the manureTypeStored and speciesGroup parameters of manureStore. The manureType and name parameters of theManure are set to manureTypeStored and the manure name respectively.

The C and N transformations in housing are calculated in Housing.cs public void DoHousing() and the manure is transferred to the manure storage. The transformations in manure storage are calculated in ManureStore.cs public void DoManurestore(). The UpdateManureExchange function in ManureStore.cs takes a copy of theManure object in the manure storage and calls the theManureExchange.AddToManureExchange function in GlobalVars.cs. GlobalVars.cs in its theManureExchangeClass maintains an array of manure class called manuresStored. The AddToManureExchange function checks to see if there is already an element in the manuresStored array corresponding to the type to be added. It does so by comparing speciesGroup and manureType parameters of the incoming manure with the corresponding parameters of each of the manures already in the array. If a manure of the same type exists in the array, the amount of additional manure (donor manure) is added to the amount of the existing manure (recipient manure). If no such manure is found, a new manure is added to the array and the amount equated to the amount of the donor manure.

*When finished with housing and degasing in the manurestores then the amount of manure that is left in List<manure> manuresStored and List<manure> manuresProduced.*

Applying manure to crops:

One can apply 0 to many manures using Manure\_applied-tag under each crop. The information about the manure is saved in manureApplied in the crop class. Each record contains a ManureType and a SpeciesGroup. The valid ManureType values are shown in Table 3. Manure applications are made in the crop class function public void CalculateManureInput(bool lockit). If locket is false (which is the case when the function is called for the first time for a given crop), it then calls the GlobalVars function theManureExchange.TakeManure(amountTotalN, lengthOfSequence, ManureType, speciesGroup). This function then looks to see if there is any manure of this type (i.e. same speciesGroup (Table 2) and manure type (Table 3) in the GlobalVars.cs theManureExchangeClass manuresStored array. If this is the case, the amount to be applied to the crop is removed from the manure in the GlobalVars.cs theManureExchangeClass manuresStored array. If there is no manure of the correct type or all the manure of this type available on the farm has been used, the function looks through the theManureExchangeClass array of manure called manureImported to see if this manure type has already been imported by this or a previous crop. If it has, the amount imported is increased accordingly. If not, a new manure is added to the manureImported array. The properties of the manure are obtained by searching fertMan.xml for the correct manure. The manure is correct for manureType < 12 if the manureType and speciesGroup parameters both agree. For manureType=> 12 (mixed species manures), only the manureType must be correct.

*public void CalculateManureInput(bool lockit) will either import manure or taking in from List<manure> manuresStored. Default values for imported manure will be rea from the fertAndManure-file based upon ManureType, speciesGroup and zone. Information about imported manure is stored in manuresImported.*

theManureExchangeClass-output

The theManureExchangeClass contains three arrays; the manureStored, the importedManure and the manureProduced. At the end of the simulation, the manureProduced array contains information about the amounts of each manure produced on the farm, the manureStored array contains information on the amount of manure of each type that remains unapplied on the farm and so must be exported and the importedManure contains the amount of manure of each type imported.

theManureExchangeClass tag in the output file contains 3 sub tags describing the amounts and characteristics of each of the manures in the above arrays. The exportedManure-tag contains information about manure exported, the producedManure about the manure produced and the: importedManurea bout manure imported.

NOT USED?

Appendix:

|  |  |
| --- | --- |
| Manure Type | Description |
| 1 | Slurry |
| 2 | Solid manure |
| 3 | deep litter |
| 4 | Liquid manure |
| 5 | solid fraction |
| 6 | liquid fraction |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 | Degassed biomasse |
| 13 | Acidified slurry |