Model: hr-inclNOxExtendedWithFilters

October 17, 2020

Contents

1 Model

1.1 End

1.2 Total

This process summarizes the contribution of the individual Modules to the total NH3 emission from a farm.

1.2.1 Outputs

nh3_nanimalproduction Annual NH3 emission from farm.

```
Val(nh3_napplication, Application) +
Val(nh3_nstorage, Storage) +
Val(nh3_nlivestock, Livestock)
```

nh3_ntotal Annual NH3 emission from farm.

```
Val(nh3_napplication, Application) +
Val(nh3_nstorage, Storage) +
Val(nh3_nlivestock, Livestock) +
Val(nh3_nplantproduction, PlantProduction)
```

1.3 PlantProduction

This process summarizes the contribution of the plant production to the total NH3 emission.

1.3.1 Differences to DYNAMO

1.3.2 Outputs

nh3_nplantproduction Annual NH3 emission from plant production.

```
Val(nh3_nmineralfertiliser, PlantProduction::MineralFertiliser) +
Val(nh3_nrecyclingfertiliser, PlantProduction::RecyclingFertiliser)
```

compost Amount of compost in t per year.

```
Val(compost, PlantProduction::RecyclingFertiliser);
```

solid_digestate Amount of Solid digestate in t per year.

```
Val(solid_digestate, PlantProduction::RecyclingFertiliser);
```

liquid_digestate Amount of liquid digestate in m3 per year.

```
Val(liquid_digestate, PlantProduction::RecyclingFertiliser);
```

1.4 PlantProduction::RecyclingFertiliser

This process computes the annual average NH3 emission from recycling fertiliser application. The standard emission factor is corrected according to the TAN content and the application rate, the application technology, the crops to which the manure is applied, respected soft measures during application and the application season, the time lag between application and incorporation, the incorporation technology used, and the proportion of solid manure that is applied on hot days.

1.4.1 References:

Vanderweerden and Jarvis (1997)

1.4.2 Inputs

compost Amount of compost (in t fresh matter per year). Kompost besteht aus Grünabfällen nicht-landwirtschafltlicher Herkunft von gewerblich-industriellen Anlagen oder von Feldrand-kompostierung.

solid_digestate Amount of solid digestate form anaerobic digestion plants

liquid_digestate Amount of liquid digestate form anaerobic digestion plants

1.4.3 Outputs

nh3_ncompost NH3 emission from compost.

```
In(compost) * Tech(er_compost);
```

nh3_nsolid_degestate NH3 emission from solid digestate.

```
In(solid_digestate) * Tech(er_solid_digestate);
```

nh3_nliquid_degestate NH3 emission from liquid digestate.

```
In(liquid_digestate) * Tech(er_liquid_digestate);
```

nh3_nrecyclingfertiliser NH3 emission from total recycling fertiliser.

```
In(liquid_digestate) * Tech(er_liquid_digestate) +
In(solid_digestate) * Tech(er_solid_digestate) +
In(compost) * Tech(er_compost);
```

compost Amount of compost in t /a.

```
In(compost);
```

solid_digestate Amount of Solid digestate in t /a.

```
In(solid_digestate);
```

liquid_digestate Amount of liquid digestate in m3 /a.

```
In(liquid_digestate);
```

1.5 PlantProduction::MineralFertiliser

This process computes the annual average NH3 emission from mineral fertiliser application.

1.5.1 References:

Qiao, C.L., Liu, L.L., Hu, S.J., Compton, J.E., Greaver, T.L., Li, Q.L. 2015. How inhibiting nitrification affects nitrogen cycle and reduces environmental impacts of anthropogenic nitrogen input. Global Change Biol. 21(3): 1249-1257.

Pan, B.B., Lam, S.K., Mosier, A., Luo, Y.Q., Chen, D.L. 2016. Ammonia volatilization from synthetic fertilizers and its mitigation strategies: A global synthesis. Agric. Ecosyst. Environ. 232: 283-289.

1.5.2 Inputs

soil_ph Soil pH value

mineral_fertiliser_ammoniumNitrate_amount Amount of ammonium nitrate in kg fertilizer/year.

mineral_fertiliser_ammoniumNitrate_N_content N content of ammonium nitrate in percent

mineral_fertiliser_calciumAmmoniumNitrate_amount Amount of calcium ammonium nitrate in kg fertilizer/year.

mineral_fertiliser_calciumAmmoniumNitrate_N_content N content of calcium ammonium nitrate in percent

mineral_fertiliser_ammoniumSulphate_amount Amount of ammonium sulphate in kg fertilizer/year.

mineral_fertiliser_ammoniumSulphate_N_content N content of ammomiun sulphate in percent

mineral_fertiliser_urea_amount Amount of urea in kg fertilizer/year.

mineral_fertiliser_urea_N_content N content of urea in percent

mineral_fertiliser_sulfamid_amount Amount of sulfamid in kg fertilizer/year.

mineral_fertiliser_sulfamid_N_content N content of sulfamid in percent

mineral_fertiliser_calciumNitrate_amount Amount of calcium nitrate (Kalksalpeter) in kg fertilizer/year.

mineral_fertiliser_calciumNitrate_N_content N content of calcium nitrate (Kalksalpeter) in percent

mineral_fertiliser_calciumCyanamid_amount Amount of calcium cyanamid in kg fertilizer/year.

mineral_fertiliser_calciumCyanamid_N_content N content of calcium cyanamid in percent

mineral_fertiliser_entec_amount Amount of entec in kg fertilizer/year.

mineral_fertiliser_entec_N_content N content of Entec in percent

mineral_fertiliser_np_amount Amount of NP mixtures in kg fertilizer/year.

mineral_fertiliser_np_N_content N content of NP mixtures in percent

mineral_fertiliser_nk_amount Amount of NK mixtures in kg fertilizer/year.

mineral_fertiliser_nk_N_content N content of NK mixtures in percent

mineral_fertiliser_npk_amount Amount of NPK mixtures in kg fertilizer/year.

mineral_fertiliser_npk_N_content N content of NPK mixtures in percent

mineral_fertiliser_entec2_amount Amount of Entec2 in kg fertilizer/year.

mineral_fertiliser_entec2_N_content N content of Entec2 in percent

mineral_fertiliser_other_amount Amount of other mineral fertilizers in kg fertilizer/year.

mineral_fertiliser_other_N_content N content of other mineral fertilizers in percent

1.5.3 Outputs

nh3_n_mineral_fertiliser_ammoniumNitrate NH3 emission from mineral fertiliser application from ammonium nitrate.

```
my $nAmount = In(mineral_fertiliser_ammoniumNitrate_amount) * In(mineral_fertiliser_ammoniumNitrate_N_content
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_ammoniumNitrate_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_calciumAmmoniumNitrate NH3 emission from mineral fertiliser application from calcium ammonium nitrate.

```
my $nAmount = In(mineral_fertiliser_calciumAmmoniumNitrate_amount) * In(mineral_fertiliser_calciumAmmoniumNit
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_calciumAmmoniumNitrate_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_ammoniumSulphate NH3 emission from mineral fertiliser application from ammonium sulphate.

```
my $nAmount = In(mineral_fertiliser_ammoniumSulphate_amount) * In(mineral_fertiliser_ammoniumSulphate_N_conte
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_ammoniumSulphate_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_urea NH3 emission from mineral fertiliser application from urea.

```
my $nAmount = In(mineral_fertiliser_urea_amount) * In(mineral_fertiliser_urea_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_urea_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_sulfamid NH3 emission from mineral fertiliser application from sulfamid

```
my $nAmount = In(mineral_fertiliser_sulfamid_amount) * In(mineral_fertiliser_sulfamid_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_sulfamid_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_calciumNitrate NH3 emission from mineral fertiliser application from calcium nitrate (Kalksalpeter)

```
my $nAmount = In(mineral_fertiliser_calciumNitrate_amount) * In(mineral_fertiliser_calciumNitrate_N_content)/
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_calciumNitrate_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_calciumCyanamid NH3 emission from mineral fertiliser application from calcium cyanamid

```
my $nAmount = In(mineral_fertiliser_calciumCyanamid_amount) * In(mineral_fertiliser_calciumCyanamid_N_content
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_calciumCyanamid_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_entec NH3 emission from mineral fertiliser application from Entec

```
my $nAmount = In(mineral_fertiliser_entec_amount) * In(mineral_fertiliser_entec_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_entec_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_np NH3 emission from mineral fertiliser application from NP mixtures

```
my $nAmount = In(mineral_fertiliser_np_amount) * In(mineral_fertiliser_np_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_np_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_nk NH3 emission from mineral fertiliser application from NK mixtures

```
my $nAmount = In(mineral_fertiliser_nk_amount) * In(mineral_fertiliser_nk_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_nk_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_npk NH3 emission from mineral fertiliser application from NPK mixtures

```
my $nAmount = In(mineral_fertiliser_npk_amount) * In(mineral_fertiliser_npk_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_npk_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_entec2 NH3 emission from mineral fertiliser application from Entec2

```
my $nAmount = In(mineral_fertiliser_entec2_amount) * In(mineral_fertiliser_entec2_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_entec2_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_n_mineral_fertiliser_other NH3 emission from mineral fertiliser application from other mineral fertilizers

```
my $nAmount = In(mineral_fertiliser_other_amount) * In(mineral_fertiliser_other_N_content)/100;
my $ph = In(soil_ph);
my $er = $TE->{'er_mineral_fertiliser_other_' . $ph . '_pH'};
return $nAmount * $er;
```

nh3_nmineralfertiliser NH3 emission from mineral fertiliser application.

```
return
Out(nh3_n_mineral_fertiliser_ammoniumNitrate) +
Out(nh3_n_mineral_fertiliser_calciumAmmoniumNitrate) +
Out(nh3_n_mineral_fertiliser_ammoniumSulphate) +
Out(nh3_n_mineral_fertiliser_urea) +
Out(nh3_n_mineral_fertiliser_sulfamid) +
Out(nh3_n_mineral_fertiliser_calciumNitrate) +
Out(nh3_n_mineral_fertiliser_calciumCyanamid) +
Out(nh3_n_mineral_fertiliser_entec) +
Out(nh3_n_mineral_fertiliser_np) +
Out(nh3_n_mineral_fertiliser_nk) +
Out(nh3_n_mineral_fertiliser_npk) +
Out(nh3_n_mineral_fertiliser_entec2) +
Out(nh3_n_mineral_fertiliser_other);
```

1.6 Application

This process summarizes the contribution of the individual manure systems to the total NH3 emission from manure application.

1.6.1 Differences to DYNAMO

The categories "Soil absorptive" and "application before rain" are omitted since the practice is unknown and experimental results are not available (according to the decision of the steering group from 02/07/2007).

The distinction between the categories incorporation of solid manure by chisel plough or plough are omitted since the difference is unclear (according to the decision of the steering group from 02/07/2007).

The category "rapid incorporation" is replaced by "application manure" since slurry is hardly incorporated in Switzerland. The entire category is adapted to UNECE (2007) including new categories.

1.6.2 Outputs

n_into_application_liquid Annual N flux into liquid manure application.

```
Val(n_into_application_liquid, Storage);
```

n_into_application_solid Annual N flux into solid manure application.

```
Val(n_into_application_solid, Storage);
```

n_into_application Annual N flux into manure application.

```
Out(n_into_application_liquid) P+
Out(n_into_application_solid);

####################################################


####################################################
#### tan into application (solid/liquid/total)
```

tan_into_application_liquid Annual TAN flux into liquid manure application.

```
Val(tan_into_application_liquid, Storage);
```

tan_into_application_solid Annual TAN flux into solid manure the application.

```
Val(tan_into_application_solid, Storage);
```

tan_into_application Annual TAN flux into manure the application.

```
Out(tan_into_application_liquid) P+
Out(tan_into_application_solid);

####################################################



####################################################
#### NH3 losses
```

nh3_napplication_solid Annual NH3 emission from solid manure application.

```
Out(tan_into_application_solid) P*
(
Val(er_nh3_napplication_solid_no_poultry, Application::SolidManure::Solid) P+
Val(er_nh3_napplication_solid_poultry, Application::SolidManure::Poultry)
);
```

nh3_napplication_liquid Annual NH3 emission from liquid manure application.

```
Val(nh3_napplication_liquid, Application::Slurry);
```

nh3_napplication Annual NH3 emission from manure application.

```
Out(nh3_napplication_solid) P+
Out(nh3_napplication_liquid);

####################################################



####################################################
#### NxOx losses

## 2a) N2 losses
```

n2_napplication_liquid Annual N2 emission from liquid manure application.

```
Out(n_into_application_liquid) P*
Val(er_n2_napplication_liquid, Application::Slurry);
```

n2_napplication_solid Annual N2 emission from solid manure application.

```
Out(n_into_application_solid) P*
(
Val(er_n2_napplication_solid_no_poultry, Application::SolidManure::Solid) P+
Val(er_n2_napplication_solid_poultry, Application::SolidManure::Poultry)
);
```

n2_napplication Annual N2 emission from manure application.

```
Out(n2_napplication_liquid) P+
Out(n2_napplication_solid);


## 2b) NO losses
```

no_napplication_liquid Annual NO emission from liquid manure application.

```
Out(n_into_application_liquid) P*
Val(er_no_napplication_liquid, Application::Slurry);
```

no_napplication_solid Annual NO emission from solid manure application.

```
Out(n_into_application_solid) P*
(
Val(er_no_napplication_solid_no_poultry, Application::SolidManure::Solid) P+
Val(er_no_napplication_solid_poultry, Application::SolidManure::Poultry)
);
```

no_napplication Annual NO emission from manure application.

```
Out(no_napplication_solid) P+
Out(no_napplication_liquid);

## 2c) N2O losses
```

n2o_napplication_liquid Annual N2O emission from liquid manure application.

```
Out(n_into_application_liquid) P*
Val(er_n2o_napplication_liquid, Application::Slurry);
```

n2o_napplication_solid Annual N2O emission from solid manure application.

```
Out(n_into_application_solid) P*
(
Val(er_n2o_napplication_solid_no_poultry, Application::SolidManure::Solid) P+
Val(er_n2o_napplication_solid_poultry, Application::SolidManure::Poultry)
);
```

n2o_napplication Annual N2O emission from manure application.

```
Out(n2o_napplication_solid) P+
Out(n2o_napplication_liquid);

## 2d) NxOx losses -> Total.nhd?

####################################################


####################################################
#### n out of application (solid/liquid/total)
```

n_remain_application_liquid Annual N flux out of liquid manure application.

```
Out(n_into_application_liquid) P-
Out(nh3_napplication_liquid) P-
Out(n2_napplication_liquid)  P-
Out(no_napplication_liquid)  P-
Out(n2o_napplication_liquid);
```

n_remain_application_solid Annual N flux out of solid manure application.

```
Out(n_into_application_solid) P-
Out(nh3_napplication_solid) P-
Out(n2_napplication_solid)  P-
Out(no_napplication_solid)  P-
Out(n2o_napplication_solid);
```

n_remain_application Annual N flux out of manure application.

```
Out(n_remain_application_liquid)  P+
Out(n_remain_application_solid);

####################################################



####################################################
#### tan out of application (solid/liquid/total)
```

tan_remain_application_liquid Annual TAN flux out of liquid manure application.

```
Out(tan_into_application_liquid) P-
Out(nh3_napplication_liquid) P-
Out(n2_napplication_liquid)  P-
Out(no_napplication_liquid)  P-
Out(n2o_napplication_liquid);
```

tan_remain_application_solid Annual TAN flux out of solid manure application.

```
Out(tan_into_application_solid) P-
Out(nh3_napplication_solid) P-
Out(n2_napplication_solid)  P-
Out(no_napplication_solid)  P-
Out(n2o_napplication_solid);
```

tan_remain_application Annual TAN flux out of manure application.

```
Out(tan_remain_application_liquid)  P+
Out(tan_remain_application_solid);

####################################################
```

1.7 Application::SolidManure::Poultry

This process computes the annual average NH3 emission from poultry manure application. The standard emission factor is corrected according to the TAN content and the application rate, the application technology, the crops to which the manure is applied, respected soft measures during application and the application season, the time lag between application and incorporation, the incorporation technology used, and the proportion of solid manure that is applied on hot days.

The correction factor are based on the same input parameters as the application for solid manure.

1.7.1 References:

Menzi H, Keller M, Katz P, Fahrni M, Neftel A 1997. Ammoniakverluste nach der Anwendung von Mist. Agrarforschung 4:328-331.

Menzi H, Katz P, Frick R, Fahrni M, Keller M 1997: Ammonia emissions following the application of solid manure to grassland. In: Jarvis S.C. and Pain B.F. (Eds.): "Nitrogen Emissions from Grassland", CAB, 265-274.

TODO (Cyrill Bonjour): Add correct calculation based on Ntot and TAN!

1.7.2 Outputs

er_nh3_napplication_solid_poultry NH3 emission rate for solid manure application.

```
scale(
Val(has_poultry, ::Livestock),
Tech(er_App_manure) *
Val(c_incorp_time, Poultry::CincorpTime) *
Val(c_season, Cseason) *
Val(c_free_factor_application_solidmanure, CfreeFactor)
);
```

er_n2_napplication_solid_poultry N2 emission from solid manure application from poultry.

```
scale(
Val(has_poultry, ::Livestock),
Tech(er_n2_App_manure)
);
```

er_no_napplication_solid_poultry NO emission from solid manure application from poultry.

```
scale(
Val(has_poultry, ::Livestock),
Tech(er_no_App_manure)
);
```

er_n2o_napplication_solid_poultry N2O emission from solid manure application from poultry.

```
scale(
Val(has_poultry, ::Livestock),
Tech(er_n2o_App_manure)
);
```

1.8 Application::SolidManure::Poultry::CincorpTime

This process computes the correction factor for the time lag between application and incorporation of the poultry manure.

1.8.1 References:

Menzi H, Keller M, Katz P, Fahrni M, Neftel A 1997. Ammoniakverluste nach der Anwendung von Mist. Agrarforschung 4:328-331.

Menzi H, Katz P, Frick R, Fahrni M, Keller M 1997: Ammonia emissions following the application of solid manure to grassland. In: Jarvis S.C. and Pain B.F. (Eds.): "Nitrogen Emissions from Grassland", CAB, 265-274.

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.8.2 Outputs

c_incorp_time Correction factor taking into account the time lag between application and incorporation of the solid manure.

```
return  1 + ( Val(incorp_lw1h,..::CincorpTime) * Tech(eff_inc_lw1h) +
Val(incorp_lw4h,..::CincorpTime) * Tech(eff_inc_lw4h) +
Val(incorp_lw8h,..::CincorpTime) * Tech(eff_inc_lw8h) +
Val(incorp_lw1d,..::CincorpTime) * Tech(eff_inc_lw1d) +
Val(incorp_lw3d,..::CincorpTime) * Tech(eff_inc_lw3d) +
Val(incorp_gt3d,..::CincorpTime) * Tech(eff_inc_gt3d) +
```

```
Val(incorp_none,..::CincorpTime) * Tech(eff_inc_none)
);
```

1.9 Application::SolidManure::Solid

This process computes the annual average NH3 emission from solid manure application (liquid/solid and deep litter). The standard emission factor is corrected according to the TAN content and the application rate, the application technology, the crops to which the manure is applied, respected soft measures during application and the application season, the time lag between application and incorporation, the incorporation technology used, and the proportion of solid manure that is applied on hot days.

1.9.1 References:

Menzi H, Keller M, Katz P, Fahrni M, Neftel A 1997. Ammoniakverluste nach der Anwendung von Mist. Agrarforschung 4:328-331.

Menzi H, Katz P, Frick R, Fahrni M, Keller M 1997: Ammonia emissions following the application of solid manure to grassland. In: Jarvis S.C. and Pain B.F. (Eds.): "Nitrogen Emissions from Grassland", CAB, 265-274.

Webb, J., Sommer, S.G., Kupper, T., Groenestein, C.M., Hutchings, N., Eurich-Menden, B., Rodhe, L., Misselbrook, T., Amon, B. 2012. Emissions of ammonia, nitrous oxide and methane during the management of solid manures. A review. In: Lichtfouse, E., (eds.). Agroecology and Strategies for Climate Change. Heidelberg, Germany: Springer-Verlag GmbH. pp 67-108.

1.9.2 Outputs

er_nh3_napplication_solid_no_poultry NH3 emission rates for solid manure application from all animal categories except poultry.

```
scale(
# er cattle
scale(
Val(has_cattle, ::Livestock),
Tech(er_App_manure_dairycows_cattle)
) P+
# er pigs
scale(
Val(has_pigs, ::Livestock),
Tech(er_App_manure_pigs)
) P+
# er others
scale(
Val(has_others, ::Livestock),
Tech(er_App_manure_horses_otherequides_smallruminants)
),
# other factors
Val(c_incorp_time, Solid::CincorpTime) *
Val(c_season, Cseason) *
Val(c_free_factor_application_solidmanure, CfreeFactor)
);

##### er n2
```

er_n2_napplication_solid_no_poultry N2 emission rates for solid manure application from all animal categories except poultry.

```
scale(
Val(has_no_poultry, ::Livestock),
Tech(er_n2_App_manure)
);

##### er no
```

er_no_napplication_solid_no_poultry N2 emission rates for solid manure application from all animal categories except poultry.

```
scale(
Val(has_no_poultry, ::Livestock),
Tech(er_no_App_manure)
);

##### er n2o
```

er_n2o_napplication_solid_no_poultry N2 emission rates for solid manure application from all animal categories except poultry.

```
scale(
Val(has_no_poultry, ::Livestock),
Tech(er_n2o_App_manure)
);
```

1.10 Application::SolidManure::CfreeFactor

1.10.1 Inputs

free_correction_factor Factor to define free ?

1.10.2 Outputs

c_free_factor_application_solidmanure Free reduction of the Emission rate for the Yard.

```
if(In(free_correction_factor)//0 != 0) {
writeLog({en=>"You have entered an additional emission mitigation measure for a application of solid manure c
. In(free_correction_factor)
. "\%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für die Ausbringung von Mist von "
. In(free_correction_factor)
. "\% eingegeben!\n",
fr=>"Vous avez introduit une mesure supplémentaire limitant les émissions dues à l'épandage de fumier de "
. In(free_correction_factor) . "\% !\n"});
return 1 - In(free_correction_factor)/100;

} else {
return 1;
}
```

1.11 Application::SolidManure::Cseason

This process computes the correction factor for the seasons the solid manure is applied.

1.11.1 References:

Menzi H, Frick R, Kaufmann R 1997. Ammoniak-Emissionen in der Schweiz: Ausmass und technische Beurteilung des Reduktionspotentials. Eidgenössische Forschungsanstalt für Agrarökologie und Landbau, Zürich-Reckenholz. 107pp.

Frick R, Menzi H 1997. Hofdüngeranwendung: Wie Ammoniakverluste vermindern? Auch einfache Massnahmen wirken. FAT Bericht Nr. 496.

1.11.2 Inputs

appl_summer Share of solid manure applied June to August (in %).

appl_autumn_winter_spring Share of solid manure applied September to May (in %).

1.11.3 Outputs

appl_summer .

```
return (In(appl_summer)/100);
```

appl_autumn_winter_spring .

```
return (In(appl_autumn_winter_spring)/100);
```

c_season Correction factor of the standard emission rate depending on season of application.

```
if( abs(In(appl_summer)+In(appl_autumn_winter_spring)-100) < 0.000001 ){
(1 + ( Out(appl_summer) * Tech(c_summer) +
Out(appl_autumn_winter_spring) * Tech(c_autumn_winter_spring)));
}
else {
writeLog({en => "Please correct accordingly: the categories of solid manure incorporated do not add up to 100
de => "Bitte korrigieren: die Summe der Kategorien Einarbeitung von Mist ist nicht gleich 100\%!\n",
fr => "Veuillez corriger : la somme des catégories «Part de fumier incorporé» n'est pas égale à 100% !",
});
(1 + ( Out(appl_summer) * Tech(c_summer) +
Out(appl_autumn_winter_spring) * Tech(c_autumn_winter_spring)));
}
```

1.12 Application::SolidManure::Solid::CincorpTime

This process computes the correction factor for the time lag between application and incorporation of the solid manure.

1.12.1 References:

Menzi H, Keller M, Katz P, Fahrni M, Neftel A 1997. Ammoniakverluste nach der Anwendung von Mist. Agrarforschung 4:328-331.

1.12.2 Outputs

c_incorp_time Correction factor taking into account the time lag between application and incorporation of the solid manure.

```
return 1 + ( Val(incorp_lw1h,..::CincorpTime) * Tech(eff_inc_lw1h) +
Val(incorp_lw4h,..::CincorpTime) * Tech(eff_inc_lw4h) +
Val(incorp_lw8h,..::CincorpTime) * Tech(eff_inc_lw8h) +
Val(incorp_lw1d,..::CincorpTime) * Tech(eff_inc_lw1d) +
Val(incorp_lw3d,..::CincorpTime) * Tech(eff_inc_lw3d) +
Val(incorp_gt3d,..::CincorpTime) * Tech(eff_inc_gt3d) +
Val(incorp_none,..::CincorpTime) * Tech(eff_inc_none)
);
```

1.13 Application::SolidManure::CincorpTime

This process computes the correction factor for the time lag between application and incorporation of the solid manure (from all animal categories).

1.13.1 References:

Menzi H, Keller M, Katz P, Fahrni M, Neftel A 1997. Ammoniakverluste nach der Anwendung von Mist. Agrarforschung 4:328-331.

1.13.2 Inputs

incorp_lw1h Share of incorporated solid manure within 1 hour.

incorp_lw4h Share of incorporated solid manure within 4 hours.

incorp_lw8h Share of incorporated solid manure within 8 hours.

incorp_lw1d Share of incorporated solid manure within 1 day.

incorp_lw3d Share of incorporated solid manure within 3 days.

incorp_gt3d Share of incorporated solid manure after 3 days.

incorp_none Share of solid manure not incorporated.

1.13.3 Outputs

incorp_lw1h Share of incorporated solid manure within 1 hour.

```
return (In(incorp_lw1h)/100)
```

incorp_lw4h Share of incorporated solid manure within 4 hour.

```
return (In(incorp_lw4h)/100)
```

incorp_lw8h Share of incorporated solid manure within 8 hour.

```
return (In(incorp_lw8h)/100)
```

incorp_lw1d Share of incorporated solid manure within 1 day.

```
return (In(incorp_lw1d)/100)
```

incorp_lw3d Share of incorporated solid manure within 3 days.

```
return (In(incorp_lw3d)/100)
```

incorp_gt3d Share of incorporated solid manure after 3 days.

```
return (In(incorp_gt3d)/100)
```

incorp_none Share of not-incorporated solid manure.

```
return (In(incorp_none)/100)
```

test_incorp_time Correction factor taking into account the time lag between application and incorporation of the solid manure.

```
if( ( In(incorp_lw1h) +
In(incorp_lw4h) +
In(incorp_lw8h) +
In(incorp_lw1d) +
In(incorp_lw3d) +
```

```
In(incorp_gt3d) +
In(incorp_none)
) >= 99.9999
&& ( In(incorp_lw1h) +
In(incorp_lw4h) +
In(incorp_lw8h) +
In(incorp_lw1d) +
In(incorp_lw3d) +
In(incorp_gt3d) +
In(incorp_none)
) <= 100.0001
){  return 1;
}else{
writeLog({en=>"Please correct accordingly: the categories of solid manure incorporated do not add up to 100\%
de=>"Bitte korrigieren: die Summe der Kategorien Einarbeitung von Mist ist nicht gleich 100\%!",
fr=>"Veuillez corriger : la somme des catégories «Part de fumier incorporé» n'est pas égale à 100\% !"
} );
return  0;
}
```

1.14 Application::SolidManure

This process computes the annual average NH3 emission from solid manure application (liquid/solid and deep litter). The standard emission factor is corrected according to the TAN content and the application rate, the application technology, the crops to which the manure is applied, respected soft measures during application and the application season, the time lag between application and incorporation, the incorporation technology used, and the proportion of solid manure that is applied on hot days.

1.14.1 References:

Menzi H, Keller M, Katz P, Fahrni M, Neftel A 1997. Ammoniakverluste nach der Anwendung von Mist. Agrarforschung 4:328-331.

Menzi H, Katz P, Frick R, Fahrni M, Keller M 1997: Ammonia emissions following the application of solid manure to grassland. In: Jarvis S.C. and Pain B.F. (Eds.): "Nitrogen Emissions from Grassland", CAB, 265-274.

1.15 Application::Slurry

This process computes the annual NH3 emission from slurry application. The standard emission factor for slurry application is corrected according to the TAN content and the application rate, the application technology, the crops to which the manure is applied, soft measures applied during application and the application season.

Since slurry is hardly incorporated in Switzerland, no correction for incorporeation was made for slurry application.

1.15.1 References:

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

Frick R, Menzi H, Katz P 1996. Ammoniakverluste nach der Hofdüngeranwendung. FAT-Bericht Nr. 486.

Katz P E 1996. Dissertation: Ammoniakemissionen nach der Gülleanwendung auf Grünland. Diss. ETH Nr. 11382. Dissertation. Eidgenössische Technische Hochschule Zürich.

Menzi H, Frick R, Kaufmann R 1997a. Ammoniak-Emissionen in der Schweiz: Ausmass und technische Beurteilung des Reduktionspotentials. Eidgenössische Forschungsanstalt für Agrarökologie und Landbau, Zürich-Reckenholz. 107pp.

Menzi H, Katz, PE, Fahrni M, Neftel A, Frick R 1998. A simple empirical model based on regression analysis to estimate ammonia emissions after manure application. Atmospheric Environment 32:301-307.

Sogaard H T, Sommer S G, Hutchings N J, Huijsmans J F M, Bussink D W, Nicholson F 2002. Ammonia volatilization from field-applied animal slurry - the ALFAM model. Atmospheric Environment 36: 3309-3319.

Sommer S G 2001b. Effect of coposting on nutrient loss and nitrogen availability of cattle deep litter. European Journal of Agronomy 14: 123-133.

1.15.2 Outputs

pigs_share_n_out_livestock_liquid Pigs share of N flux out of livestock liquid manure

```
scalar(
multiplyPairwise(
Val(has_pigs, ::Livestock),
Val(n_out_livestock_liquid, ::Livestock)
)
) / scalar(Val(n_out_livestock_liquid, ::Livestock));
```

er_App_liquid Emission rate for slurry application based on TAN of the slurry. The average rate has been derived from Sommer (2001b), Sogaard et al. (2002), Menzi et al (1998), Menzi et al. (1997a) and Sogaard et al. (2002)

```
Out(pigs_share_n_out_livestock_liquid) *
Tech(er_App_pigs_liquid) +
(1 - Out(pigs_share_n_out_livestock_liquid)) *
Tech(er_App_cattle_liquid);
```

nh3_napplication_liquid Total annual NH3 emission from slurry application.

```
scale(
Val(tan_into_application_liquid, ::Storage),
Val(c_tech, Slurry::Ctech) *
Val(c_soft, Slurry::Csoft) *
Val(c_season, Slurry::Cseason) *
Val(c_free_factor_application, Slurry::CfreeFactor) *
(
(1 - Val(share_fermented_slurry, Slurry::Cfermented)) *
Out(er_App_liquid) +
Val(share_fermented_slurry, Slurry::Cfermented) *
Tech(er_App_fermented_slurry) +
Val(c_app, Slurry::Applrate)
)
);

####################################################


####################################################
#### liquid application correct way (implement and clean up as soon as permitted)

## better (but still incorrect) way
```

pigs_share_tan_out_livestock_liquid Pigs share of TAN flux out of livestock liquid manure

```
scalar(
multiplyPairwise(
Val(has_pigs, ::Livestock),
Val(tan_out_livestock_liquid, ::Livestock)
)
) / scalar(Val(tan_out_livestock_liquid, ::Livestock));
```

er_App_liquid_better Emission rate for slurry application based on TAN of the slurry. The average rate has been derived from Sommer (2001b), Sogaard et al. (2002), Menzi et al (1998), Menzi et al. (1997a) and Sogaard et al. (2002)

```
Out(pigs_share_tan_out_livestock_liquid) *
Tech(er_App_pigs_liquid) +
(1 - Out(pigs_share_tan_out_livestock_liquid)) *
Tech(er_App_cattle_liquid);
```

nh3_napplication_liquid_better Total annual NH3 emission from slurry application.

```
scale(
Val(tan_into_application_liquid, ::Storage),
Val(c_tech, Slurry::Ctech) *
Val(c_soft, Slurry::Csoft) *
Val(c_season, Slurry::Cseason) *
Val(c_free_factor_application, Slurry::CfreeFactor) *
(
(1 - Val(share_fermented_slurry, Slurry::Cfermented)) *
Out(er_App_liquid_better) +
Val(share_fermented_slurry, Slurry::Cfermented) *
Tech(er_App_fermented_slurry) +
Val(c_app, Slurry::Applrate)
)
);


## correct total flow & detail report based on actual flow into application
```

nh3_napplication_liquid_correct Total annual NH3 emission from slurry application.

```
scale(
# only pigs
scale(
multiplyPairwise(
Val(has_pigs, ::Livestock),
Val(tan_into_application_liquid, ::Storage)
),
(1 - Val(share_fermented_slurry, Slurry::Cfermented)) *
Tech(er_App_pigs_liquid)
) P+
# only cattle
scale(
multiplyPairwise(
Val(has_cattle, ::Livestock),
Val(tan_into_application_liquid, ::Storage)
),
(1 - Val(share_fermented_slurry, Slurry::Cfermented)) *
Tech(er_App_cattle_liquid)
) P+
# both
scale(
Val(tan_into_application_liquid, ::Storage),
Val(share_fermented_slurry, Slurry::Cfermented) *
Tech(er_App_fermented_slurry) +
Val(c_app, Slurry::Applrate)
),
# other factors affecting emission
Val(c_tech, Slurry::Ctech) *
Val(c_soft, Slurry::Csoft) *
Val(c_season, Slurry::Cseason) *
```

```
Val(c_free_factor_application, Slurry::CfreeFactor)
);
```

er_nh3_application_liquid_correct Emission rate

```
Out(nh3_napplication_liquid_correct) P/
(
Val(tan_into_application_liquid, ::Storage) P+
add(
sign(Val(tan_into_application_liquid, ::Storage)),
-1
)
);

####################################################



####################################################
#### NxOx loss
```

er_n2_napplication_liquid Total annual N2 emission from slurry application.

```
scale(
Val(has_no_poultry, ::Livestock),
Tech(er_n2_App_liquid)
);
```

er_no_napplication_liquid Total annual NO emission from slurry application.

```
scale(
Val(has_no_poultry, ::Livestock),
Tech(er_no_App_liquid)
);
```

er_n2o_napplication_liquid Total annual N2O emission from slurry application.

```
scale(
Val(has_no_poultry, ::Livestock),
Tech(er_n2o_App_liquid)
);

####################################################
```

1.16 Application::Slurry::CfreeFactor

1.16.1 Inputs

free_correction_factor Factor to define free ?

1.16.2 Outputs

c_free_factor_application Free reduction of the Emission rate for the Yard.

```
if(In(free_correction_factor)//0 != 0) {
writeLog({en=>"You have entered an additional emission mitigation measure for a application of slurry of "
. In(free_correction_factor)
. "\% !\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für die Ausbringung von Gülle von "
. In(free_correction_factor)
. "\% eingegeben!\n",
fr=>"Vous avez introduit une mesure supplémentaire limitant les émissions dues à l’épandage de lisier de " .
. "\% !\n"});
return 1 - In(free_correction_factor)/100;
} else {
```

```
return 1;
}
```

1.17 Application::Slurry::Cfermented

This process computes the correction factor of fermented slurry

1.17.1 References:

1.17.2 Inputs

fermented_slurry Share of anaerobically digested slurry

1.17.3 Outputs

share_fermented_slurry Share of fermented slurry.

```
In(fermented_slurry)/100;
```

1.18 Application::Slurry::Cseason

This process computes the correction factor for the seasons the slurry is applied.

1.18.1 References:

Menzi H, Frick R, Kaufmann R 1997. Ammoniak-Emissionen in der Schweiz: Ausmass und technische Beurteilung des Reduktionspotentials. Eidgenössische Forschungsanstalt für Agrarökologie und Landbau, Zürich-Reckenholz. 107pp.

Frick R, Menzi H 1997. Hofdüngeranwendung: Wie Ammoniakverluste vermindern? Auch einfache Massnahmen wirken. FAT Bericht Nr. 496.

1.18.2 Inputs

appl_summer Share of slurry applied June to August (in %).

appl_autumn_winter_spring Share of slurry applied September to May.

1.18.3 Outputs

appl_summer .

```
return (In(appl_summer)/100);
```

appl_autumn_winter_spring .

```
return (In(appl_autumn_winter_spring)/100);
```

c_season Correction factor of the standard emission rate depending on season of application.

```
if( abs(In(appl_summer)+In(appl_autumn_winter_spring)-100) < 0.000001 ){
return (1 + ( Out(appl_summer) * Tech(c_summer) +
Out(appl_autumn_winter_spring) * Tech(c_autumn_winter_spring)));
}
else {
writeLog({en => "Please correct accordingly: the categories of \"share of solid manure applied June to August
. "and \"share of solid manure applied September to May\" do not add up to 100\%!\n",
de => "Bitte korrigieren: die Summe der Kategorien Ausbringung von Gülle im Sommer und von "
. "September bis Mai ist nicht gleich 100\%\n",
fr => "Veuillez corriger: la somme des catégories «Part de lisier épandu en été» et «de "
. "septembre à mai» n'est pas égale à 100% !\n"
});
# Warning!
return (1 + ( Out(appl_summer) * Tech(c_summer) +
Out(appl_autumn_winter_spring) * Tech(c_autumn_winter_spring)));
}
```

1.19 Application::Slurry::Csoft

This process computes the correction factor if different soft measures for slurry application are respected.

1.19.1 References:

Menzi H, Frick R, Kaufmann R 1997. Ammoniak-Emissionen in der Schweiz: Ausmass und technische Beurteilung des Reduktionspotentials. Eidgenössische Forschungsanstalt für Agrarökologie und Landbau, Zürich-Reckenholz. 107pp.

Frick R, Menzi H 1997. Hofdüngeranwendung: Wie Ammoniakverluste vermindern? Auch einfache Massnahmen wirken. FAT Bericht Nr. 496.

1.19.2 Inputs

appl_evening Share of slurry applied in the evening after 18:00.

appl_hotdays Proportion of slurry applied on hot days.

1.19.3 Outputs

c_hotdays Correction factor of the emission rate if slurry is applied on hot days.

```
return $TE->{'c_hotdays_'.In(appl_hotdays)};
```

c_soft Correction factor of the emission rate if slurry is applied by considering different kinds of "soft measures".

```
return ( 1 + (( In(appl_evening)/100) * Tech(c_evening) + Out(c_hotdays)));
```

1.20 Application::Slurry::Applrate

This process computes the correction factor as a function of the application rate and the TAN content of the slurry. The equation has been described by Menzi et al. (1998). The correction factor is calculated based on the slurry application rate per ha and the TAN content of the slurry compared to the emission rate occurring with a standard application rate of 30 m3 and a TAN content of 1.15 kg N / m3.

1.20.1 References:

Menzi H, Katz, P E, Fahrni M, Neftel A, Frick R 1998. A simple empirical model based on regression analysis to estimate ammonia emissions after manure application. Atmospheric Environment 32: 301-307.

TODO (Haral Menzi): Please confirm, as description was not finished by Beat Reidy, appl_rate and tan was not included in calculation.

1.20.2 Inputs

dilution_parts_water Specific slurry dilution. TAN contents have been calculated based on a standard dilution of 1:1 with a TAN content of 1.15 kg N /m3.

appl_rate Application rate, mean volume of slurry applied on a ha per deployment.

1.20.3 Outputs

TAN_content TAN content of the slurry compared to the emission rate occuring with a standard application rate of 30 m3 and a TAN content of 1.15 kg N / m3.

```
2.3*(1/(In(dilution_parts_water)+1));
```

c_app Correction factor taking into account the slurry application rate per ha and the TAN content of the slurry.

```
if( (In(appl_rate) * Out(TAN_content)) > 0 ){
( ( ( 19.41 * Out(TAN_content) + 4.2 * 1.102 - 9.51) *
(In(appl_rate) * 0.0214 + 0.36)) /
(In(appl_rate) * Out(TAN_content)) - Tech(norm_er));
}else{ return 1; }
```

1.21 Application::Slurry::Ctech

This process computes the correction factor according to the technology used for the slurry application.

1.21.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. UNECE executive body for the convention on long-range transboundary air pollution, Working Group on Strategies and Review, EB.AIR/WH.5/2007/13/ 16 July 2007, Genf. 35p.

Frick R, Menzi H 1997. Hofdüngeranwendung: Wie Ammoniakverluste vermindern? Auch einfache Massnahmen wirken. FAT Bericht 496.

Menzi H, Frick R, Kaufmann R 1997. Ammoniak-Emissionen in der Schweiz: Ausmass und technische Beurteilung des Reduktionspotentials. Eidgenössische Forschungsanstalt für Agrarökologie und Landbau, Zürich-Reckenholz. 107pp.

1.21.2 Inputs

share_splash_plate Share of slurry applied with splash plate.

share_trailing_hose Share of slurry applied with trailing hose.

share_trailing_shoe Share of slurry applied with trailing shoes.

share_shallow_injection Share of slurry applied with shallow injection.

share_deep_injection Share of slurry applied with deep injection.

1.21.3 Outputs

share_deep_injection Share

```
return (In(share_deep_injection)/100);
```

share_shallow_injection Share

```
return (In(share_shallow_injection)/100);
```

share_trailing_shoe Share

```
return (In(share_trailing_shoe)/100);
```

share_trailing_hose Share

```
return (In(share_trailing_hose)/100);
```

share_splash_plate Share

```
return (In(share_splash_plate)/100);
```

c_tech Reduction factor for the emission due to the used application technology as compared to broadcasting.

```
if( abs( Out(share_deep_injection) +
Out(share_shallow_injection) +
Out(share_trailing_shoe) +
Out(share_trailing_hose) +
Out(share_splash_plate)
- 1)  < 1e-8 )
{

return 1 + ( Out(share_deep_injection) * Tech(red_deep_injection) +
Out(share_shallow_injection) * Tech(red_shallow_injection) +
Out(share_trailing_shoe) * Tech(red_trailing_shoe) +
Out(share_trailing_hose) * Tech(red_trailing_hose) +
Out(share_splash_plate) * Tech(red_splash_plate)
);
}
else{
writeLog({en => "Please correct accordingly: the categories categories of slurry application do not add up to
de => "Bitte korrigieren: die Summe der Kategorien Gülleausbringung ist nicht gleich 100\%!\n",
fr => "Veuillez corriger : la somme des catégories «Epandage du lisier» n’est pas égale à 100% !\%!\n"});
# Warning + Test!
return 1 + ( Out(share_deep_injection) * Tech(red_deep_injection) +
Out(share_shallow_injection) * Tech(red_shallow_injection) +
Out(share_trailing_shoe) * Tech(red_trailing_shoe) +
Out(share_trailing_hose) * Tech(red_trailing_hose) +
Out(share_splash_plate) * Tech(red_splash_plate)
);
}
```

1.22 Storage

This process calculates the NH3 emission from slurry storage, considering both slurry from slurry based systems and liquid from liquid/solid systems. The surface to volume ration (measure for the emitting surface), the cover type and artificial slurry aeration are accounted for via correction factors. Calculations are performed independently for slurry and liquid from liquid/solid systems with the same procedure.

1.22.1 References:

De Bode MJC, 1990. Vergleich der Ammoniakemissionen aus verschiedenen Flüssigmistlagersystemen. In: Ammoniak in der Umwelt. Hrsg.: KTBL und VDI, Münster, 34:1-13.

1.22.2 Outputs

has_liquid_storage True (1) if a Liquid Storage (Volume) is present.

```
my $volume = Sum(volume, Storage::Slurry) // 0;
if ( $volume > 0){
return 1;
}
else {
if( Val(n_out_livestock_liquid, ::Livestock) > 0 ) {
writeLog(
{
en => "No storage for slurry defined although slurry is produced!\n",
de => "Es ist kein Güllelager eingegeben, obwohl Gülle anfällt!\n",
fr => "Aucun stock de lisier n’est mentionné, alors qu’il y a production de lisier!\n"
}
);
}
return 0;
}



####################################################
##### mineralization liquid
```

mineralization_liquid_cattle Annual TAN mineralized from not TAN fraction in liquid storage.

```
(
Val(n_out_livestock_liquid, ::Livestock) P-
Val(tan_out_livestock_liquid, ::Livestock)
) P*
scale(
Val(has_cattle, ::Livestock),
Tech(mineralizationrate_liquid)
);
```

mineralization_liquid_pigs Annual TAN mineralized from not TAN fraction in liquid storage.

```
(
Val(n_out_livestock_liquid, ::Livestock) P-
Val(tan_out_livestock_liquid, ::Livestock)
) P*
scale(
Val(has_pigs, ::Livestock),
Tech(mineralizationrate_liquid)
);
```

mineralization_liquid Annual TAN mineralized from not TAN fraction in liquid storage.

```
Out(mineralization_liquid_cattle) P+
Out(mineralization_liquid_pigs);

####################################################
```

```
##################################################
##### n into storage
```

n_into_storage_liquid Annual N flux into liquid storage.

```
Val(n_out_livestock_liquid, ::Livestock);
```

n_into_storage_solid Annual N flux into liquid storage.

```
Val(n_out_livestock_solid, ::Livestock) P*
(
Val(share_into_storage_solid_no_poultry, Storage::SolidManure::Solid) P+
Val(share_into_storage_solid_poultry, Storage::SolidManure::Poultry)
);
```

n_into_storage Annual N flux into liquid storage.

```
Out(n_into_storage_liquid) P+
Out(n_into_storage_solid);

##################################################
```

```
##################################################
##### tan into storage
```

tan_into_storage_liquid Annual N flux into liquid storage.

```
Val(tan_out_livestock_liquid, ::Livestock);
```

tan_into_storage_solid Annual N flux into liquid storage.

```
Val(tan_out_livestock_solid, ::Livestock) P*
(
Val(share_into_storage_solid_no_poultry, Storage::SolidManure::Solid) P+
Val(share_into_storage_solid_poultry, Storage::SolidManure::Poultry)
);
```

tan_into_storage_liquid_pigs Annual N flux into liquid storage.

```
multiplyPairwise(
Val(has_pigs, ::Livestock),
Out(tan_into_storage_liquid)
);
```

tan_into_storage_liquid_cattle Annual N flux into liquid storage.

```
multiplyPairwise(
Val(has_cattle, ::Livestock),
Out(tan_into_storage_liquid)
);

##################################################
```

```
##################################################
##### nh3 loss
```

maxlimit_of_nh3_nstorage_liquid_pigs Upper limit of the annual NH3 emission from liquid storage of pig slurry.

```
scalar(
multiplyPairwise(
Val(has_pigs, ::Livestock),
Val(tan_out_livestock_liquid, ::Livestock) P+
Out(mineralization_liquid) P-
Val(nxox_nliquid, ::Livestock)
)
);
```

```
# eventually change to scaling by tan!
```

nh3_nstorage_liquid_pigs Annual NH3 emission from liquid storage of pig slurry.

```
my $nh3_loss_pigs = Sum(nh3_ntank_liquid_pigs, Storage::Slurry);
if ( scalar($nh3_loss_pigs) <  scalar(Out(maxlimit_of_nh3_nstorage_liquid_pigs)) ) {
scale(
Out(tan_into_storage_liquid_pigs),
$nh3_loss_pigs /
scalar(Out(tan_into_storage_liquid_pigs))
);
} else {
writeLog(
{
en => "NH3 loss from pigs is limited to maximum available TAN!\n",
de => "TODO\n",
fr => "TODO\n"
}
);
Out(maxlimit_of_nh3_nstorage_liquid_pigs);
}
```

maxlimit_of_nh3_nstorage_liquid_cattle Upper limit of the annual NH3 emission from liquid storage of pig slurry.

```
multiplyPairwise(
Val(has_cattle, ::Livestock),
Val(tan_out_livestock_liquid, ::Livestock) P+
Out(mineralization_liquid) P-
Val(nxox_nliquid, ::Livestock)
);

# eventually change to scaling by tan!
```

nh3_nstorage_liquid_cattle Annual NH3 emission from liquid storage of cattle slurry.

```
my $nh3_loss_cattle = Sum(nh3_ntank_liquid_cattle, Storage::Slurry);
if ( scalar($nh3_loss_cattle) <  scalar(Out(maxlimit_of_nh3_nstorage_liquid_cattle)) ) {
scale(
Out(tan_into_storage_liquid_cattle),
$nh3_loss_cattle /
scalar(Out(tan_into_storage_liquid_cattle))
);
} else {
writeLog(
{
en => "NH3 loss from cattle is limited to maximum available TAN!\n",
de => "TODO\n",
fr => "TODO\n"
}
);
Out(maxlimit_of_nh3_nstorage_liquid_cattle);
}
```

nh3_nstorage_liquid Annual NH3 emission from storage.

```
Out(nh3_nstorage_liquid_pigs) P+
Out(nh3_nstorage_liquid_cattle);
```

nh3_nstorage_solid Annual NH3 emission from storage.

```
Out(tan_into_storage_solid) P*
(
Val(er_nh3_nstorage_solid_no_poultry, Storage::SolidManure::Solid) P+
Val(er_nh3_nstorage_solid_poultry, Storage::SolidManure::Poultry)
);
```

nh3_nstorage Annual NH3 emission from storage.

```
Out(nh3_nstorage_liquid) P+
Out(nh3_nstorage_solid);

####################################################



####################################################
##### n/tan into application liquid
## immob eigentlich Netto 40% von TAN zu Norg -> zu Beginn (ohne Abzug NH3)
## dafür NH3 basierend auf (TANin + TANout) / 2 * EF
```

immobilization Annual TAN immobilized from TAN fraction in solid manure storage.

```
(
Out(tan_into_storage_solid) P-
Out(nh3_nstorage_solid)
) P*
(
Val(immobilization_rate_no_poultry, Storage::SolidManure::Solid) P+
Val(immobilization_rate_poultry, Storage::SolidManure::Poultry)
);

####################################################



####################################################
##### n/tan into application liquid
```

n_into_application_liquid Annual N flux out of storage for application.

```
Val(n_out_livestock_liquid, ::Livestock) P-
Out(nh3_nstorage_liquid) P-
Val(nxox_nliquid, ::Livestock);
```

tan_into_application_liquid Annual N flux as TAN out of storage for application.

```
Val(tan_out_livestock_liquid, ::Livestock) P-
Out(nh3_nstorage_liquid) P-
Val(nxox_nliquid, ::Livestock) P+
Out(mineralization_liquid);


####################################################


####################################################
##### n/tan into application solid
```

n_into_application_solid Annual N flux out of storage for manure application.

```
Val(n_out_livestock_solid, ::Livestock) P-
Val(nxox_nsolid, ::Livestock) P-
Out(nh3_nstorage_solid);
```

tan_into_application_solid Annual TAN flux out of storage for manure application.

```
Val(tan_out_livestock_solid, ::Livestock) P-
Val(nxox_nsolid, ::Livestock) P-
Out(nh3_nstorage_solid) P-
Out(immobilization);

####################################################


####################################################
##### n/tan into application (move later to application?)
```

n_into_application Annual N flux out of storage for application.

```
Out(n_into_application_liquid) P+
Out(n_into_application_solid) ;
```

tan_into_application Annual TAN flux out of storage for application.

```
Out(tan_into_application_liquid) P+
Out(tan_into_application_solid);

####################################################

######
# possibly add:
# - n/tan_directly_applied_solid
# - n/tan_out_storage*
```

1.23 Storage::Slurry

This Process calculates the annual NH3 emission from a single liquid manuare storage, considering a specific emission factor.

1.23.1 References

De Bode MJC, 1990. Vergleich der Ammoniakemissionen aus verschiedenen Flüssigmistlagersystemen. In: Ammoniak in der Umwelt. Hrsg.: KTBL und VDI, Münster, 34:1-13. Menzi H, Frick R, Kaufmann R, 1997a. Ammoniak-Emissionen in der Schweiz: Ausmass und technische Beurteilung des Reduktionspotentials. Eidgenössische Forschungsanstalt für Agrarökologie und Landbau, Zürich-Reckenholz. 107pp. Sommer SG, Christensen BT, Nielsen NE, Schjorring JK, 1993. Ammonia volatilization during storage of cattle and pig slurry - effect of surface cover. Journal of Agricultural Science 121:63-71.

1.23.2 Inputs

volume Volume of slurry store.

depth Depth of slurry store.

mixing_frequency Frequency of mixing of slurry store.

1.23.3 Outputs

c_mixing Correction factor for number of mixing frequency in storage, accoording to selected levels.

```
my $mixing = In(mixing_frequency) // '7_to_12_times_per_year';
return $TE->{'c_mixing_' . $mixing};
```

depth Depth of slurry storage.

```
In(depth)
```

volume Volume of slurry storage.

```
In(volume)
```

surface_area Surface area of slurry storage.

```
if (not defined In(depth) or Out(depth) <= 0) {
return 0;
} else {
return Out(volume) / Out(depth);
}
```

nh3_ntank_liquid_pigs Annual NH3 emission from slurry storage.

```
Val(er_nh3_storage_liquid_pigs, Slurry::EFLiquid) *
Out(surface_area) *
Out(c_mixing) *
Val(c_free_factor_storage_slurry, Slurry::EFLiquid) ;
```

nh3_ntank_liquid_cattle Annual NH3 emission from slurry storage.

```
Val(er_nh3_storage_liquid_cattle, Slurry::EFLiquid) *
Out(surface_area) *
Out(c_mixing) *
Val(c_free_factor_storage_slurry, Slurry::EFLiquid) ;
```

1.24 Storage::Slurry::EFLiquid

1.24.1 References

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

De Bode MJC, 1990. Vergleich der Ammoniakemissionen aus verschiedenen Flüssigmistlagersystemen. In: Ammoniak in der Umwelt. Hrsg.: KTBL und VDI, Münster(D), 34:1-13.

Sommer SG, Christensen BT, Nielsen NE, Schjorring JK, 1993. Ammonia volatilization during storage of cattle and pig slurry - effect of surface cover. Journal of Agricultural Science 121:63-71.

1.24.2 Inputs

cover_type Cover type of liquid storage.

contains_cattle_manure Describes if the specific storage contains cattle manure.

contains_pig_manure Describes if the specific storage contains pig manure.

free_correction_factor Factor to define free ?

1.24.3 Outputs

er_nh3_storage_liquid_pigs Scaled emission factor of a specific liquid storage for pig slurry.

```
if (not defined In(cover_type)) {
writeLog({
en => 'FIX: Invalid values for "cover_type" (' . In(cover_type) . ')' . "\n",
de => 'FIX: Ungültige Eingabe für "cover_type" (' . In(cover_type) . ')' . "\n",
fr => 'FIX: Entrée non valable pour "cover_type" (' . In(cover_type) . ')' . "\n"
});
return 0;
}
my $er_pigs = $TE->{'ef_pig_'.In(cover_type)};
# scale er pigs
if ( lc In(contains_pig_manure) eq 'no' ) {
return 0;
} elsif ( lc In(contains_pig_manure) eq 'yes' and lc In(contains_cattle_manure) eq 'no' ) {
return $er_pigs;
```

```
} elsif ( lc In(contains_pig_manure) eq 'yes' and lc In(contains_cattle_manure) eq 'yes' ) {
return Val(n_out_livestock_liquid_pigs_share, ::Livestock) * $er_pigs;
} else {
writeLog({
en => 'Invalid values for "Contains pigs slurry" (' . In(contains_pig_manure) . ')' . "\n",
de => 'Ungültige Eingabe für "Enthält Schweinegülle" (' . In(contains_pig_manure) . ')' . "\n",
fr => 'Entrée non valable pour "Contient du lisier de porc" (' . In(contains_pig_manure) . ')' . "\n"
});
return Val(n_out_livestock_liquid_pigs_share, ::Livestock) * $er_pigs;
}
```

er_nh3_storage_liquid_cattle Scaled emission factor of a specific liquid storage for cattle slurry.

```
if (not defined In(cover_type)) {
return 0;
}
my $er_cattle = $TE->{'ef_cattle_'.In(cover_type)};
if ( lc In(contains_cattle_manure) eq 'no' ) {
return 0;
} elsif ( lc In(contains_cattle_manure) eq 'yes' and lc In(contains_pig_manure) eq 'no' ) {
return $er_cattle;
} elsif ( lc In(contains_pig_manure) eq 'yes' and lc In(contains_cattle_manure) eq 'yes' ) {
return (1 - Val(n_out_livestock_liquid_pigs_share, ::Livestock)) * $er_cattle;
} else {
writeLog({
en => 'Invalid values for "Contains cattle slurry" (' . In(contains_cattle_manure) . ')' . "\n",
de => 'Ungültige Eingabe für "Enthält Rindergülle" (' . In(contains_cattle_manure) . ')' . "\n",
fr => 'Entrée non valable pour "Contient du lisier de bovin" (' . In(contains_cattle_manure) . ')' . "\n"
});
return (1 - Val(n_out_livestock_liquid_pigs_share, ::Livestock)) * $er_cattle;
}
```

c_free_factor_storage_slurry Free reduction of the Emission rate for the Yard.

```
if(In(free_correction_factor)//0 != 0) {
writeLog({en=>"You have entered an additional emission mitigation measure for a storage of slurry of "
. In(free_correction_factor)
. "\%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für ein Güllelager von "
. In(free_correction_factor)
. "\% eingegeben!\n",
fr=>"Vous avez introduit une mesure supplémentaire limitant les émissions "
. "du stock de lisier de " . In(free_correction_factor) . "\%.\n" });
return 1 - In(free_correction_factor)/100;

} else {
return 1;
}
```

1.25 Storage::SolidManure::Poultry

This process calculates the annual NH3 emission from poultry manure storage, considering a mean emission rate on TAN flux in storage.

1.25.1 References

European Agricultural Gaseous Emissions Inventory Researchers Network - EAGER workshop, January 2008.

1.25.2 Inputs

share_applied_direct_poultry_manure Share of poultry manure applied to land without storage.

share_covered_basin Share of droppings or mist from poultry stored in covered basin.

free_correction_factor Factor to define free ?

1.25.3 Outputs

c_covered_basin Correction factor for storage droppings or mist in coverd basin.

```
1 - ( Tech(c_droppings_mist_covered_basin) * In(share_covered_basin)/100 );
```

c_free_factor_storage_poultrymanure Free reduction of the Emission rate for the Yard.

```
if(In(free_correction_factor)//0 != 0) {
writeLog({en=>"You have entered an additional emission mitigation measure for a storage of poultry manure of
. In(free_correction_factor)
. " \%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für das Geflügelmistlager von "
. In(free_correction_factor)
. "\% eingegeben!\n",
fr=> "Vous avez introduit une mesure supplémentaire limitant les émissions du stock "
. "de fumier de volaille de " . In(free_correction_factor) . "\%.\n"
});
return 1 - In(free_correction_factor)/100;

} else {
return 1;
}
```

n_check Check shares of directly applied and covered storage of poultry manure

```
if (  (In(share_applied_direct_poultry_manure) + In(share_covered_basin)) > 100  ) {
writeLog({en=>"The sum of Share of poultry manure applied to land without storage and Share of poultry manure
de=>"Die Summe von Anteil von direkt ohne Lagerung ausgebrachtem Geflügelmist und Anteil von gedeckt gelagert
fr=>"La somme de Part des fientes ou du fumier de volaille épandu directement sans stockage et de Part des fi
});
}
return;
```

share_into_storage_solid_poultry Annual TAN flux into solid storage from pigs.

```
scale(
Val(has_poultry, ::Livestock),
(1 - In(share_applied_direct_poultry_manure) / 100)
);
```

er_nh3_nstorage_solid_poultry Annual NH3 emission from poultry manure storage.

```
scale(
scale(
Val(has_poultry_LGO, ::Livestock),
Tech(er_layers_growers_other_poultry)
) P+
scale(
Val(has_poultry_TB, ::Livestock),
Tech(er_turkeys_broilers)
),
Out(c_covered_basin) *
Out(c_free_factor_storage_poultrymanure)
);
```

immobilization_rate_poultry Annual TAN immobilized from TAN fraction in solid manure storage.

```
scale(
Val(has_poultry, ::Livestock),
Tech(immobilizationrate_poultry)
);
```

1.26 Storage::SolidManure::Solid

This process calculates the annual NH3 emission from solid manure storage, considering a mean emission rate on TAN flux in solid storage.

1.26.1 Inputs

share_applied_direct_cattle_other_manure Share of cattles, equides and small ruminants manure applied to land without storage.

share_covered_basin_cattle_manure Share of droppings or mist from cattle stored in covered basin.

free_correction_factor_cattle_manure Factor to define free ?

share_applied_direct_pig_manure Share of pig manure applied to land without storage.

share_covered_basin_pig_manure Share of droppings or mist from pigs stored in covered basin.

free_correction_factor_pig_manure Factor to define free ?

1.26.2 Outputs

n_check_cattle Check shares of directly applied and covered storage of cattle manure

```
if (  (In(share_applied_direct_cattle_other_manure) + In(share_covered_basin_cattle_manure)) > 100  ) {
writeLog({en=>"The sum of share of cattle manure applied to land without storage and share of cattle manure c
de=>"Die Summe von Anteil von direkt ohne Lagerung ausgebrachtem Mist von Rindvieh und Anteil von gedeckt gel
fr=>"La somme de part du fumier de bovins épandu directement sans stockage et de part du fumier de bovins sto
});
}
return;
```

n_check_pigs Check shares of directly applied and covered storage of pig manure

```
if (  (In(share_applied_direct_pig_manure) + In(share_covered_basin_pig_manure)) > 100  ) {
writeLog({en=>"The sum of share of cattle manure applied to land without storage and share of cattle manure c
de=>"Die Summe von Anteil von direkt ohne Lagerung ausgebrachtem Schweinemist und Anteil von gedeckt gelagert
fr=>"La somme de part du fumier de porcs épandu directement sans stockage et de part du fumier de porcs stock
});
}
return;
```

c_free_factor_storage_solidmanure_cattle Free reduction of the Emission rate for the Yard Storage, cattle manure.

```
if(In(free_correction_factor_cattle_manure)//0 != 0) {
writeLog({en=>"You have entered an additional emission mitigation measure for a storage of cattle solid manur
. In(free_correction_factor_cattle_manure)
. "\%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für das Mistlager der Rinder von"
. In(free_correction_factor_cattle_manure)
. "\% eingegeben!\n",
fr=>"Vous avez introduit une mesure supplémentaire limitant les émissions du "
. "stock de fumier des porcs de " . In(free_correction_factor_cattle_manure)
. "\%.\n"
});
```

```
return 1 - In(free_correction_factor_cattle_manure)/100;

} else {
return 1;
}
```

c_free_factor_storage_solidmanure_pig Free reduction of the Emission rate for Storage Pig manure.

```
if(In(free_correction_factor_pig_manure)//0 != 0) {
writeLog({en=>"You have entered an additional emission mitigation measure for a storage of pigs solid manure
. In(free_correction_factor_cattle_manure)
. "\%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für das Mistlager der Schweine von"
. In(free_correction_factor_pig_manure)
. "\% eingegeben!\n",
fr=>"'Vous avez introduit une mesure supplémentaire limitant les émissions du "
. "stock de fumier des bovins de" . In(free_correction_factor_pig_manure)
. "\%.\n"
});
return 1 - In(free_correction_factor_pig_manure)/100;

} else {
return 1;
}
```

c_covered_basin_cattle Correction factor for manure of cattle stored in coverd basin.

```
1 - ( Tech(c_covered_basin_cattle_manure) * In(share_covered_basin_cattle_manure)/100 );
```

c_covered_basin_pig Correction factor for manure of pigs stored in coverd basin.

```
1 - ( Tech(c_covered_basin_pig_manure) * In(share_covered_basin_pig_manure)/100 );
```

share_into_storage_solid_no_poultry Annual TAN flux into solid storage from pigs.

```
scale(
Val(has_cattle, ::Livestock) P+
Val(has_others, ::Livestock),
(1 - In(share_applied_direct_cattle_other_manure) / 100)
) P+
scale(
Val(has_pigs, ::Livestock),
(1 - In(share_applied_direct_pig_manure) / 100)
);
```

er_nh3_nstorage_solid_no_poultry Annual NH3 emission from solid storage.

```
scale(
Val(has_cattle, ::Livestock) P+
Val(has_others, ::Livestock),
Tech(er_tan_cattle_other) *
Out(c_covered_basin_cattle) *
Out(c_free_factor_storage_solidmanure_cattle)
) P+
scale(
Val(has_pigs, ::Livestock),
Tech(er_tan_pigs) *
Out(c_covered_basin_pig) *
Out(c_free_factor_storage_solidmanure_pig)
);
```

immobilization_rate_no_poultry Annual TAN immobilized from TAN fraction in solid manure storage.

```
scale(
Val(has_no_poultry, ::Livestock),
Tech(immobilizationrate_solid)
);
```

1.27 Livestock

This process summarizes the annual NH3 emission from livestock (housing, yard and grazing) for all animal categories. Further it calculates the N flux into storage from housing and yard. The manure is splited in solid and liquid/slurry.

1.27.1 Outputs

n_into_housing Total annual N excreted by all animals.

```
Sum(n_into_housing, Livestock::OtherCattle::Housing) P+
Sum(n_into_housing, Livestock::DairyCow::Housing) P+
Sum(n_into_housing, Livestock::Pig::Housing) P+
Sum(n_into_housing, Livestock::FatteningPigs::Housing) P+
Sum(n_into_housing, Livestock::Equides::Housing) P+
Sum(n_into_housing, Livestock::SmallRuminants::Housing) P+
Sum(n_into_housing, Livestock::RoughageConsuming::Housing) P+
Sum(n_into_housing, Livestock::Poultry::Housing);
```

n_into_yard Total annual N excreted by all animals.

```
Sum(n_into_yard, Livestock::OtherCattle::Yard) P+
Sum(n_into_yard, Livestock::DairyCow::Yard) P+
Sum(n_into_yard, Livestock::Equides::Yard);
```

n_into_grazing Total annual N excreted by all animals.

```
Sum(n_into_grazing, Livestock::OtherCattle::Grazing) P+
Sum(n_into_grazing, Livestock::DairyCow::Grazing) P+
Sum(n_into_grazing, Livestock::Pig::Grazing) P+
Sum(n_into_grazing, Livestock::FatteningPigs::Grazing) P+
Sum(n_into_grazing, Livestock::Equides::Grazing) P+
Sum(n_into_grazing, Livestock::SmallRuminants::Grazing) P+
Sum(n_into_grazing, Livestock::RoughageConsuming::Grazing) P+
Sum(n_into_outdoor, Livestock::Poultry::Outdoor);
```

n_into_livestock Total annual N excreted by all animals.

```
Out(n_into_housing) P+
Out(n_into_yard);


####################################################
##### tan input -> Clean up labels etc!!!
```

tan_into_housing Total annual N excreted by all animals.

```
Sum(tan_into_housing, Livestock::OtherCattle::Housing) P+
Sum(tan_into_housing, Livestock::DairyCow::Housing) P+
Sum(tan_into_housing, Livestock::Pig::Housing) P+
Sum(tan_into_housing, Livestock::FatteningPigs::Housing) P+
Sum(tan_into_housing, Livestock::Equides::Housing) P+
Sum(tan_into_housing, Livestock::SmallRuminants::Housing) P+
Sum(tan_into_housing, Livestock::RoughageConsuming::Housing) P+
Sum(tan_into_housing, Livestock::Poultry::Housing);
```

tan_into_yard Total annual N excreted by all animals.

```
Sum(tan_into_yard, Livestock::OtherCattle::Yard) P+
Sum(tan_into_yard, Livestock::DairyCow::Yard) P+
Sum(tan_into_yard, Livestock::Equides::Yard);
```

tan_into_grazing Total annual N excreted by all animals.

```
Sum(tan_into_grazing, Livestock::OtherCattle::Grazing) P+
Sum(tan_into_grazing, Livestock::DairyCow::Grazing) P+
Sum(tan_into_grazing, Livestock::Pig::Grazing) P+
```

```
Sum(tan_into_grazing, Livestock::FatteningPigs::Grazing) P+
Sum(tan_into_grazing, Livestock::Equides::Grazing) P+
Sum(tan_into_grazing, Livestock::SmallRuminants::Grazing) P+
Sum(tan_into_grazing, Livestock::RoughageConsuming::Grazing) P+
Sum(tan_into_outdoor, Livestock::Poultry::Outdoor);
```

tan_into_livestock Total annual TAN excreted by all animals.

```
Out(tan_into_housing) P+
Out(tan_into_yard);

####################################################



####################################################
##### Prepare animal master categories
```

has_cattle Animal categories belonging to mastercategory _cattle_ (dairy cows and other cattle). Hash entry of 1, if N flux into livestock > 0. Hash entry of 0, if N flux into livestock = 0.

```
sign(
Sum(n_excretion, Livestock::OtherCattle::Excretion) P+
Sum(n_excretion, Livestock::DairyCow::Excretion)
);
```

has_pigs Animal categories belonging to mastercategory _pigs_ (fattening pigs and other pigs). Hash entry of 1, if N flux into livestock > 0. Hash entry of 0, if N flux into livestock = 0.

```
sign(
Sum(n_excretion, Livestock::Pig::Excretion) P+
Sum(n_excretion, Livestock::FatteningPigs::Excretion)
);
```

has_others Animal categories belonging to mastercategory _others_ (equides, small ruminants and roughage consuming). Hash entry of 1, if N flux into livestock > 0. Hash entry of 0, if N flux into livestock = 0.

```
sign(
Sum(n_excretion, Livestock::Equides::Excretion) P+
Sum(n_excretion, Livestock::SmallRuminants::Excretion) P+
Sum(n_excretion, Livestock::RoughageConsuming::Excretion)
);
```

has_poultry_LGO Animal categories belonging to mastercategory *poultry_LGO* (layers, growers and other poultry). Hash entry of 1, if N flux into livestock > 0. Hash entry of 0, if N flux into livestock = 0.

```
sign( Sum(n_excretion_layers_growers_other_poultry,Livestock::Poultry) );
```

has_poultry_TB Animal categories belonging to mastercategory *poultry_TB* (turkeys and broilers). Hash entry of 1, if N flux into livestock > 0. Hash entry of 0, if N flux into livestock = 0.

```
sign(
Sum(n_excretion_turkeys_broilers,Livestock::Poultry)
);
```

has_no_poultry Animal categories belonging to mastercategory *no_poultry* (all except poultry). Hash entry of 1, if N flux into livestock > 0. Hash entry of 0, if N flux into livestock = 0.

```
Out(has_pigs) P+
Out(has_cattle) P+
Out(has_others);
```

has_poultry Animal categories belonging to mastercategory _poultry_ (all poultry). Hash entry of 1, if N flux into livestock > 0. Hash entry of 0, if N flux into livestock = 0.

```
Out(has_poultry_LGO) P+
Out(has_poultry_TB);

#####################################################



#####################################################
##### n out housing
```

n_out_housing_liquid Annual N flux (liquid share) out of housing.

```
Sum(n_outhousing_liquid, Livestock::Pig::Housing) P+
Sum(n_outhousing_liquid, Livestock::FatteningPigs::Housing) P+
Sum(n_outhousing_liquid, Livestock::OtherCattle::Housing) P+
Sum(n_outhousing_liquid, Livestock::DairyCow::Housing);
```

n_out_housing_solid Annual N flux (liquid share) out of housing.

```
Sum(n_outhousing_solid, Livestock::Pig::Housing) P+
Sum(n_outhousing_solid, Livestock::FatteningPigs::Housing) P+
Sum(n_outhousing_solid, Livestock::OtherCattle::Housing) P+
Sum(n_outhousing_solid, Livestock::DairyCow::Housing) P+
Sum(n_outhousing_solid, Livestock::Equides::Housing) P+
Sum(n_outhousing_solid, Livestock::SmallRuminants::Housing) P+
Sum(n_outhousing_solid, Livestock::RoughageConsuming::Housing) P+
Sum(n_outhousing_solid, Livestock::Poultry::Housing);

#####################################################



#####################################################
##### n out yard
```

n_out_yard_liquid Annual N flux (liquid share) out of yard.

```
Sum(n_outyard_liquid, Livestock::OtherCattle::Yard) P+
Sum(n_outyard_liquid, Livestock::DairyCow::Yard);
```

n_out_yard_solid Annual N flux (liquid share) out of yard.

```
Sum(n_outyard_solid, Livestock::Equides::Yard);

#####################################################



#####################################################
##### n out livestock (from housing and yard into storage + directly applied)
```

n_out_livestock_liquid Annual N flux (liquid share) from housing and yard into the storage from all animal besides poultry.

```
Out(n_out_housing_liquid) P+
Out(n_out_yard_liquid);

# eventually use 'tan_out_livestock_liquid_pigs_share' for liquid storage scaling!
```

n_out_livestock_liquid_pigs_share Scalar. Share of annual liquid N flux from housing and yard into the storage from pigs (scaled by total flux).

```
(
scalar(Sum(n_liquid_from_pig,Livestock::Pig)) +
scalar(Sum(n_outhousing_liquid, Livestock::FatteningPigs::Housing))
) /
scalar(Out(n_out_livestock_liquid));
```

n_out_livestock_solid Annual N flux (solid share) from housing and yard into the storage from all animals besides poultry.

```
Out(n_out_housing_solid) P+
Out(n_out_yard_solid);
```

n_out_livestock Annual N flux (liquid and solid share) from housing and yard into the storage from all animals.

```
Out(n_out_livestock_liquid) P+
Out(n_out_livestock_solid);

####################################################



####################################################
##### n out livestock (into storage + directly applied)
```

tan_out_housing_liquid Annual N flux (liquid share) out of housing.

```
Sum(tan_outhousing_liquid, Livestock::Pig::Housing) P+
Sum(tan_outhousing_liquid, Livestock::FatteningPigs::Housing) P+
Sum(tan_outhousing_liquid, Livestock::OtherCattle::Housing) P+
Sum(tan_outhousing_liquid, Livestock::DairyCow::Housing);
```

tan_out_housing_solid Annual N flux (solid share) as TAN from housing and yard into the storage from all animals besides poultry.

```
Sum(tan_outhousing_solid, Livestock::OtherCattle::Housing) P+
Sum(tan_outhousing_solid, Livestock::DairyCow::Housing) P+
Sum(tan_outhousing_solid, Livestock::Pig::Housing) P+
Sum(tan_outhousing_solid, Livestock::FatteningPigs::Housing) P+
Sum(tan_outhousing_solid, Livestock::Equides::Housing) P+
Sum(tan_outhousing_solid, Livestock::SmallRuminants::Housing) P+
Sum(tan_outhousing_solid, Livestock::RoughageConsuming::Housing) P+
Sum(tan_outhousing_solid, Livestock::Poultry::Housing);

####################################################



####################################################
##### n out livestock (into storage + directly applied)
```

tan_out_yard_liquid Annual N flux (liquid share) out of yard.

```
Sum(tan_outyard_liquid, Livestock::OtherCattle::Yard) P+
Sum(tan_outyard_liquid, Livestock::DairyCow::Yard);
```

tan_out_yard_solid Annual N flux (solid share) as TAN from yard and yard into the storage from all animals besides poultry.

```
Sum(tan_outyard_solid, Livestock::Equides::Yard);

####################################################



####################################################
##### tan out livestock (into storage + directly applied)
```

tan_out_livestock_liquid Annual N flux (liquid share) as TAN from housing and yard into the storage from all animals besides poultry.

```
Out(tan_out_housing_liquid) P+
Out(tan_out_yard_liquid);

# eventually use for liquid storage scaling!
```

tan_out_livestock_liquid_pigs_share Scalar. Share of annual liquid N flux as TAN from housing and yard into the storage from pigs (scaled by total flux).

```
(
scalar(Sum(tan_outhousing_liquid, Livestock::Pig::Housing)) +
scalar(Sum(tan_outhousing_liquid, Livestock::FatteningPigs::Housing))
) /
scalar(Out(tan_out_livestock_liquid));
```

tan_out_livestock_solid Annual N flux (solid share) as TAN from housing and yard into the storage from all animals besides poultry.

```
Out(tan_out_housing_solid) P+
Out(tan_out_yard_solid);
```

tan_out_livestock Annual TAN flux as TAN from housing and yard into the storage from all animals besides poultry.

```
Out(tan_out_livestock_liquid) P+
Out(tan_out_livestock_solid);

####################################################



####################################################
#### nxox loss solid (Housing + Storage?!) -> rename to nxox_housing_storage + split to n2,...
```

nxox_nsolid_dairycows_cattle Annual NxOx flux (solid share) as TAN from housing and yard into the storage from dairycow and cattle.

```
Sum(n2_nsolid,Livestock::OtherCattle) P+
Sum(no_nsolid,Livestock::OtherCattle) P+
Sum(n2o_nsolid,Livestock::OtherCattle) P+
Sum(n2_nsolid,Livestock::DairyCow) P+
Sum(no_nsolid,Livestock::DairyCow) P+
Sum(n2o_nsolid,Livestock::DairyCow);
```

nxox_nsolid_equides_other Annual NxOx flux (solid share) as TAN from housing and yard into the storage from all other animals

```
Sum(n2_nsolid,Livestock::SmallRuminants) P+
Sum(n2_nsolid,Livestock::RoughageConsuming) P+
Sum(n2_nsolid,Livestock::Equides) P+

Sum(no_nsolid,Livestock::SmallRuminants) P+
Sum(no_nsolid,Livestock::RoughageConsuming) P+
Sum(no_nsolid,Livestock::Equides) P+

Sum(n2o_nsolid,Livestock::SmallRuminants) P+
Sum(n2o_nsolid,Livestock::RoughageConsuming) P+
Sum(n2o_nsolid,Livestock::Equides);
```

nxox_nsolid_poultry Annual NxOx flux (solid share) as TAN from housing and yard into the storage from poultry.

```
Sum(n2_npoultry,Livestock::Poultry) P+
Sum(no_npoultry,Livestock::Poultry) P+
Sum(n2o_npoultry,Livestock::Poultry);
```

nxox_nsolid_layers_growers_other_poultry Annual NxOx flux (solid share) as TAN from housing and yard into the storage from poultry if layer , growers or other poultry.

```
Sum(n2_npoultry_layers_growers_other_poultry,Livestock::Poultry) P+
Sum(no_npoultry_layers_growers_other_poultry,Livestock::Poultry) P+
Sum(n2o_npoultry_layers_growers_other_poultry,Livestock::Poultry);
```

nxox_nsolid_turkeys_broilers Annual NxOx flux (solid share) as TAN from housing and yard into the storage from poultry if turkeys or broilers.

```
Sum(n2_npoultry_turkeys_broilers,Livestock::Poultry) P+
Sum(no_npoultry_turkeys_broilers,Livestock::Poultry) P+
Sum(n2o_npoultry_turkeys_broilers,Livestock::Poultry);
```

nxox_nsolid_pigs Annual NxOx flux (solid share) as TAN from housing and yard into the storage from all pigs.

```
Sum(n2_nsolid,Livestock::Pig) P+
Sum(no_nsolid_pig,Livestock::Pig) P+
Sum(n2o_nsolid,Livestock::Pig) P+

Sum(n2_nsolid,Livestock::FatteningPigs) P+
Sum(no_nsolid_fattening_pig,Livestock::FatteningPigs) P+
Sum(n2o_nsolid,Livestock::FatteningPigs);
```

nxox_nsolid Annual N flux (solid share) as TAN from housing and yard into the storage from all animals besides poultry.

```
Out(nxox_nsolid_dairycows_cattle) P+
Out(nxox_nsolid_pigs) P+
Out(nxox_nsolid_equides_other) P+
Out(nxox_nsolid_poultry);

####################################################
##### nxox liquid -> rename to nxox_housing_storage + split to n2,...
```

nxox_nliquid Annual N flux (solid share) as TAN from housing and yard into the storage from all animals besides poultry.

```
Sum(n2_nliquid,Livestock::OtherCattle) P+
Sum(no_nliquid,Livestock::OtherCattle) P+
Sum(n2o_nliquid,Livestock::OtherCattle) P+

Sum(n2_nliquid,Livestock::DairyCow) P+
Sum(no_nliquid,Livestock::DairyCow) P+
Sum(n2o_nliquid,Livestock::DairyCow) P+

Sum(n2_nliquid,Livestock::Pig) P+
Sum(no_nliquid_pig,Livestock::Pig) P+
Sum(n2o_nliquid,Livestock::Pig) P+

Sum(n2_nliquid,Livestock::FatteningPigs) P+
Sum(no_nliquid_fattening_pig,Livestock::FatteningPigs) P+
Sum(n2o_nliquid,Livestock::FatteningPigs);

####################################################
```

nxox_n Annual N flux (solid share) as TAN from housing and yard into the storage from all animals besides poultry.

```
Out(nxox_nliquid) +
Out(nxox_nsolid_dairycows_cattle) +
Out(nxox_nsolid_pigs) +
Out(nxox_nsolid_equides_other) +
Out(nxox_nsolid_poultry);
```

n2_nlivestock Annual N2 emission from livestock from all animals.

```
Sum(n2_nsolid,Livestock::OtherCattle) P+
Sum(n2_nliquid,Livestock::OtherCattle) P+
Sum(n2_nsolid,Livestock::DairyCow) P+
Sum(n2_nliquid,Livestock::DairyCow) P+
Sum(n2_nsolid,Livestock::SmallRuminants) P+
Sum(n2_nsolid,Livestock::RoughageConsuming) P+
Sum(n2_nsolid,Livestock::Pig) P+
Sum(n2_nliquid,Livestock::Pig) P+
Sum(n2_nsolid,Livestock::FatteningPigs) P+
Sum(n2_nliquid,Livestock::FatteningPigs) P+
```

```
Sum(n2_nsolid,Livestock::Equides) P+
Sum(n2_npoultry,Livestock::Poultry);
```

no_nlivestock Annual NO emission from livestock from all animals.

```
Sum(no_nsolid,Livestock::OtherCattle) P+
Sum(no_nliquid,Livestock::OtherCattle) P+
Sum(no_nsolid,Livestock::DairyCow) P+
Sum(no_nliquid,Livestock::DairyCow) P+
Sum(no_nsolid,Livestock::SmallRuminants) P+
Sum(no_nsolid,Livestock::RoughageConsuming) P+
Sum(no_nsolid_pig,Livestock::Pig) P+
Sum(no_nliquid_pig,Livestock::Pig) P+
Sum(no_nsolid_fattening_pig,Livestock::FatteningPigs) P+
Sum(no_nliquid_fattening_pig,Livestock::FatteningPigs) P+
Sum(no_nsolid,Livestock::Equides) P+
Sum(no_npoultry,Livestock::Poultry);
```

n2o_nlivestock Annual N2O emission from livestock from all animals.

```
Sum(n2o_nsolid,Livestock::OtherCattle) P+
Sum(n2o_nliquid,Livestock::OtherCattle) P+
Sum(n2o_nsolid,Livestock::DairyCow) P+
Sum(n2o_nliquid,Livestock::DairyCow) P+
Sum(n2o_nsolid,Livestock::SmallRuminants) P+
Sum(n2o_nsolid,Livestock::RoughageConsuming) P+
Sum(n2o_nsolid,Livestock::Pig) P+
Sum(n2o_nliquid,Livestock::Pig) P+
Sum(n2o_nsolid,Livestock::FatteningPigs) P+
Sum(n2o_nliquid,Livestock::FatteningPigs) P+
Sum(n2o_nsolid,Livestock::Equides) P+
Sum(n2o_npoultry,Livestock::Poultry);
```

nh3_ngrazing Annual NH3 emission from all grazing areas and Poultry outdoor activities.

```
Sum(nh3_ngrazing,Livestock::OtherCattle::Grazing) P+
Sum(nh3_ngrazing,Livestock::DairyCow::Grazing) P+
Sum(nh3_ngrazing,Livestock::Pig::Grazing) P+
Sum(nh3_ngrazing,Livestock::FatteningPigs::Grazing) P+
Sum(nh3_ngrazing,Livestock::Equides::Grazing) P+
Sum(nh3_ngrazing,Livestock::SmallRuminants::Grazing) P+
Sum(nh3_ngrazing,Livestock::RoughageConsuming::Grazing) P+
Sum(nh3_free_range,Livestock::Poultry::Outdoor);
```

n2_ngrazing Annual NH3 emission from all grazing areas and Poultry outdoor activities.

```
Sum(n2_ngrazing,Livestock::OtherCattle::Grazing) P+
Sum(n2_ngrazing,Livestock::DairyCow::Grazing) P+
Sum(n2_ngrazing,Livestock::Pig::Grazing) P+
Sum(n2_ngrazing,Livestock::FatteningPigs::Grazing) P+
Sum(n2_ngrazing,Livestock::Equides::Grazing) P+
Sum(n2_ngrazing,Livestock::SmallRuminants::Grazing) P+
Sum(n2_ngrazing,Livestock::RoughageConsuming::Grazing) P+
Sum(n2_free_range,Livestock::Poultry::Outdoor);
```

no_ngrazing Annual NH3 emission from all grazing areas and Poultry outdoor activities.

```
Sum(no_ngrazing,Livestock::OtherCattle::Grazing) P+
Sum(no_ngrazing,Livestock::DairyCow::Grazing) P+
Sum(no_ngrazing,Livestock::Pig::Grazing) P+
Sum(no_ngrazing,Livestock::FatteningPigs::Grazing) P+
Sum(no_ngrazing,Livestock::Equides::Grazing) P+
Sum(no_ngrazing,Livestock::SmallRuminants::Grazing) P+
Sum(no_ngrazing,Livestock::RoughageConsuming::Grazing) P+
Sum(no_free_range,Livestock::Poultry::Outdoor);
```

n2o_ngrazing Annual NH3 emission from all grazing areas and Poultry outdoor activities.

```
Sum(n2o_ngrazing,Livestock::OtherCattle::Grazing) P+
Sum(n2o_ngrazing,Livestock::DairyCow::Grazing) P+
```

```
Sum(n2o_ngrazing,Livestock::Pig::Grazing) P+
Sum(n2o_ngrazing,Livestock::FatteningPigs::Grazing) P+
Sum(n2o_ngrazing,Livestock::Equides::Grazing) P+
Sum(n2o_ngrazing,Livestock::SmallRuminants::Grazing) P+
Sum(n2o_ngrazing,Livestock::RoughageConsuming::Grazing) P+
Sum(n2o_free_range,Livestock::Poultry::Outdoor);
```

nh3_nhousing Annual NH3 emission from all housings.

```
Sum(nh3_nhousing,Livestock::OtherCattle::Housing) P+
Sum(nh3_nhousing,Livestock::DairyCow::Housing) P+
Sum(nh3_nhousing,Livestock::SmallRuminants::Housing) P+
Sum(nh3_nhousing,Livestock::RoughageConsuming::Housing) P+
Sum(nh3_nhousing,Livestock::Pig::Housing) P+
Sum(nh3_nhousing,Livestock::FatteningPigs::Housing) P+
Sum(nh3_nhousing,Livestock::Equides::Housing) P+
Sum(nh3_nhousing,Livestock::Poultry::Housing);
```

nh3_nyard Annual NH3 emission from all yards.

```
Sum(nh3_nyard,Livestock::OtherCattle::Yard) P+
Sum(nh3_nyard,Livestock::Equides::Yard) P+
Sum(nh3_nyard,Livestock::DairyCow::Yard);

# is this variable really necessary?
```

nh3_nhousing_and_yard Annual NH3 emission from all housings and yards.

```
Out(nh3_nhousing) P+
Out(nh3_nyard);
```

n_remain_grazing Annual N remaining on pasture from all grazing areas.

```
Sum(n_remain_grazing, Livestock::OtherCattle::Grazing) P+
Sum(n_remain_grazing, Livestock::DairyCow::Grazing) P+
Sum(n_remain_grazing, Livestock::Pig::Grazing) P+
Sum(n_remain_grazing, Livestock::FatteningPigs::Grazing) P+
Sum(n_remain_grazing, Livestock::Equides::Grazing) P+
Sum(n_remain_grazing, Livestock::SmallRuminants::Grazing) P+
Sum(n_remain_grazing, Livestock::RoughageConsuming::Grazing) P+
Sum(n_remain_outdoor, Livestock::Poultry::Outdoor);
```

tan_to_grazing Annual N remaining on pasture from all grazing areas.

```
Sum(tan_into_grazing,Livestock::OtherCattle::Grazing) P+
Sum(tan_into_grazing,Livestock::DairyCow::Grazing) P+
Sum(tan_into_grazing,Livestock::Pig::Grazing) P+
Sum(tan_into_grazing,Livestock::FatteningPigs::Grazing) P+
Sum(tan_into_grazing,Livestock::Equides::Grazing) P+
Sum(tan_into_grazing,Livestock::SmallRuminants::Grazing) P+
Sum(tan_into_grazing,Livestock::RoughageConsuming::Grazing) P+
Sum(tan_into_outdoor,Livestock::Poultry::Outdoor);
```

nh3_nlivestock Annual NH3 emission from livestock from all animals.

```
Out(nh3_nhousing_and_yard) P+
Out(nh3_ngrazing);
```

1.28 Livestock::RoughageConsuming

Summarizes the annual emission of NH3 for other roughage consuming animals housing, yard and grazing. **TODO (Cyrill Bonjour):** Give a better description.

1.28.1 Outputs

n_excretion Annual N excreted by other animals.

```
Val(n_excretion, RoughageConsuming::Excretion);
```

n_from_other Annual N flux from other animals housing.

```
Val(n_outhousing, RoughageConsuming::Housing);
```

tan Annual N flux as TAN from other animals housing.

```
Val(tan_outhousing, RoughageConsuming::Housing);
```

n_liquid_from_other Annual N flux from other animals housing, liquid fraction.

```
Val(n_outhousing_liquid, RoughageConsuming::Housing);
```

tan_liquid Annual N flux as TAN from other animals housing, liquid fraction.

```
Val(tan_outhousing_liquid, RoughageConsuming::Housing);
```

n_solid_from_other Annual N flux from other animals housing, solid fraction.

```
Val(n_outhousing_solid, RoughageConsuming::Housing);
```

tan_solid Annual N flux as TAN from other animals housing, solid fraction.

```
Val(tan_outhousing_solid, RoughageConsuming::Housing);
```

nh3_nother Annual NH3 emission from other animals housing and grazing.

```
Val(nh3_nhousing, RoughageConsuming::Housing) +
Val(nh3_ngrazing, RoughageConsuming::Grazing);
```

n2_nsolid Annual N2 emission from other animals housing and grazing.

```
Val(n_into_housing, RoughageConsuming::Housing) * Tech(er_n2_nsolid);
```

no_nsolid Annual NO emission from other animals housing and grazing.

```
Val(n_into_housing, RoughageConsuming::Housing) * Tech(er_no_nsolid);
```

n2o_nsolid Annual N2O emission from other animals housing and grazing.

```
Val(n_into_housing, RoughageConsuming::Housing) * Tech(er_n2o_nsolid);
```

1.29 Livestock::RoughageConsuming::Housing

This process calculates the NH3 emission in small ruminants housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emissions, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.29.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.29.2 Outputs

n_into_housing Annual N flux into the house.

```
if( Val(n_excretion, Excretion) > Val(n_into_grazing, Grazing)  ) {
Val(n_excretion, Excretion)
- Val(n_into_grazing, Grazing);
} else { return 0; }
```

tan_into_housing Annual N flux as TAN into the house.

```
if( Val(tan_excretion, Excretion) > Val(tan_into_grazing, Grazing) ) {
Val(tan_excretion, Excretion)
- Val(tan_into_grazing, Grazing);
} else { return 0; }
```

nh3_nhousing Annual NH3 emission from small ruminants housing systems per animal place.

```
Out(tan_into_housing)
* Val(c_grazing, Housing::KGrazing)
* Tech(er_housing)
* Val(c_free_factor_housing, Housing::CFreeFactor) ;
```

n_outhousing Annual N flux out of the housing.

```
if( (Out(n_into_housing) - Out(nh3_nhousing)) > 0) {
Out(n_into_housing) - Out(nh3_nhousing);
} else { return 0; }
```

tan_outhousing Annual N flux as TAN out of the housing.

```
if( (Out(tan_into_housing) - Out(nh3_nhousing)) > 0) {
Out(tan_into_housing) - Out(nh3_nhousing);
} else { return 0; }
```

n_outhousing_liquid Annual N flux out of housing, slurry or liquid fraction of N flux.

```
0
```

tan_outhousing_liquid Annual N flux as TAN out of housing, slurry or liquid fraction of N flux.

```
0
```

n_outhousing_solid Annual N flux out of housing, manure fraction of N flux.

```
Out(n_outhousing);
```

tan_outhousing_solid Annual N flux as TAN out of housing, manure fraction of N flux.

```
Out(tan_outhousing);
```

1.30 Livestock::RoughageConsuming::Housing::CFreeFactor

TODO

1.30.1 Inputs

free_correction_factor Factor to define free.

1.30.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a housing of the "
. "category other roughage consuming of " . In(free_correction_factor)
. "\%!\n",
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Stall der "
```

```
. "Kategorie andere Raufutter Verzehrer von " . In(free_correction_factor)
. "\% eingegeben!\n",
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions dans les "
."stabulations pour 'other roughage consuming' " . In(free_correction_factor)
. "\%.\n"
});
return 1 - In(free_correction_factor) / 100;
}
else {
return 1;
}
```

1.31 Livestock::RoughageConsuming::Housing::KGrazing

This process calculates the correction factor for the non proportional change of the housing emission according to the grazing hours per day. Source is Phillips et al. (1998) and Gilhespy et al. (2006) empirical estimation was done by Thomas Kupper.

1.31.1 Outputs

c_grazing The correction factor for the reduction of the housing emission depending on the number of grazing hours per day and the grazing days per year.

```
my $k_grazing = Tech(k_grazing_a) * exp(Tech(k_grazing_b) * Val(grazing_hours, ..::Grazing));
# scale with ratio grazing_days per year
($k_grazing - 1) * Val(grazing_days, ..::Grazing) / 365 + 1;
```

1.32 Livestock::RoughageConsuming::Grazing

This process calculates the annual NH3 emission from grazing goats, fattening sheep and milk-sheep based on the total N excreted on the pastures. The N excreted is calculated according to the time the animals spend on the pasture. It assumes that the excretions on the pasture are proportional to the time the animals are grazed.

1.32.1 References:

Bussink DW 1992. Ammonia volatilization from grassland receiving nitrogen fertilizer and rotationally grazed by dairy cattle. Fertilizer Research 33:257-265.

Bussink DW 1994. Relationship between ammonia volatilization and nitrogen fertilizer application rate, intake and excretion of herbage nitrogen by cattle on grazed swards. Fertilizer Research 38:111-121.

Jarvis SC, Hatch DJ, Lockyer DR 1989. Ammonia fluxes from grazed grassland: annual losses from cattle production systems and their relation to nitrogen inputs. Journal of Agricultural Science 113:99-108.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Ross CA, Jarvis SC 2001. Measurement of emission and deposition pattern of ammonia from urine in grass swards. Atmospheric environment 35:867-875.

1.32.2 Inputs

grazing_days Average grazing days per year.

grazing_hours Average grazing hours per day.

1.32.3 Outputs

grazing_hours Grazing hours per day.

```
In(grazing_hours);
```

grazing_days Grazing days per year.

```
In(grazing_days);
```

n_into_grazing Annual N excretion during grazing for roughage consuming animals.

```
Val(n_excretion,Excretion) *
In(grazing_days)  / 365 *
In(grazing_hours) /  24;
```

tan_into_grazing Annual solable N (TAN) excretion during grazing for roughage consuming animals.

```
Val(tan_excretion,Excretion) *
In(grazing_days)  / 365 *
In(grazing_hours) /  24;
```

ef_nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
Tech(er_roughage_consuming_grazing);
```

nh3_ngrazing Annual NH3 emission from roughage consuming animals from grazing.

```
Out(tan_into_grazing) * Out(ef_nh3_ngrazing);
```

n2_ngrazing Annual total N2 emission from all grazing roughage consuming animals..

```
Out(n_into_grazing) * Tech(er_n2_roughage_consuming_grazing);
```

no_ngrazing Annual total N2 emission from all grazing roughage consuming animals..

```
Out(n_into_grazing) * Tech(er_no_roughage_consuming_grazing);
```

n2o_ngrazing Annual total N2O emission from all grazing roughage consuming animals.

```
Out(n_into_grazing) * Tech(er_n2o_roughage_consuming_grazing);
```

n_remain_grazing Annual N input on pastures.

```
Out(n_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

tan_remain_grazing Annual N input on pastures.

```
Out(tan_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

1.33 Livestock::RoughageConsuming::Excretion

This process calculated the annual N excretion of small ruminats. The standard N excretion was taken from the official Swiss fertilizer guidelines. The Nsol content of the excreta is based on e.g. Peterson et al. (1998) or Burgos et al. (2005). These values were compiled on the basis of official feeding recommendations (Flisch et al. 2009) by H. Menzi.

1.33.1 References:

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.33.2 Inputs

animalcategory Animal category (Fattening sheep, milksheep and goats).

animals Number of other animals for the selected type in barn.

inp_n_excretion Annual standard N excretion for a dairy cow

1.33.3 Outputs

animals Number of small ruminants for the selected type in barn.

```
In(animals);
```

animalcategory Animal category

```
In(animalcategory);
```

n_excretion_animal Annual standard N excretion for specified other animal category according to Flisch et al. (2009).

```
my $cat = Out(animalcategory);
my $key = 'standard_N_excretion_' . $cat;
my $exc = $TE->{$key};
$exc = In(inp_n_excretion) unless lc In(inp_n_excretion) eq 'standard';
if ( ($exc < 0.25 * $TE->{$key}) or ($exc > 1.75 * $TE->{$key})) {
writeLog({
en => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
de => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
fr => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
});
}
return $exc;
```

n_excretion Annual N excreted by a specified number of small ruminants.

```
Out(n_excretion_animal) * Out(animals);
```

tan_excretion Annual soluble N excreted by a specified number of small ruminants.

```
$TE->{'share_Nsol_'.Out(animalcategory)} * Out(n_excretion);
```

1.34 Livestock::SmallRuminants

Summarizes the annual emission of NH3 for goats, fattening sheep and milksheep from housing, yard and grazing. Summarizes the annual N flux from housing to the storage for goats, fattening sheep and milksheep.

TODO (Cyrill Bonjour): Give a better description.

1.34.1 Outputs

n_excretion Annual N excreted by other animals.

```
Val(n_excretion, SmallRuminants::Excretion);
```

n_from_other Annual N flux from other animals housing.

```
Val(n_outhousing, SmallRuminants::Housing);
```

tan Annual N flux as TAN from other animals housing.

```
Val(tan_outhousing, SmallRuminants::Housing);
```

n_liquid_from_other Annual N flux from other animals housing, liquid fraction.

```
Val(n_outhousing_liquid, SmallRuminants::Housing);
```

tan_liquid Annual N flux as TAN from other animals housing, liquid fraction.

```
Val(tan_outhousing_liquid, SmallRuminants::Housing);
```

n_solid_from_other Annual N flux from other animals housing, solid fraction.

```
Val(n_outhousing_solid, SmallRuminants::Housing);
```

tan_solid Annual N flux as TAN from other animals housing, solid fraction.

```
Val(tan_outhousing_solid, SmallRuminants::Housing);
```

nh3_nother Annual NH3 emission from other animals housing and grazing.

```
Val(nh3_nhousing, SmallRuminants::Housing) +
Val(nh3_ngrazing, SmallRuminants::Grazing);
```

n2_nsolid Annual N2 emission from other animals housing and grazing.

```
Val(n_into_housing, SmallRuminants::Housing) * Tech(er_n2_nsolid);
```

no_nsolid Annual NO emission from other animals housing and grazing.

```
Val(n_into_housing, SmallRuminants::Housing) * Tech(er_no_nsolid);
```

n2o_nsolid Annual N2O emission from other animals housing and grazing.

```
Val(n_into_housing, SmallRuminants::Housing) * Tech(er_n2o_nsolid);
```

1.35 Livestock::SmallRuminants::Housing

This process calculates the NH3 emission in small ruminants housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emissions, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.35.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.35.2 Outputs

n_into_housing Annual N flux into the house.

```
if( Val(n_excretion, Excretion) > Val(n_into_grazing, Grazing)  ) {
Val(n_excretion, Excretion)
- Val(n_into_grazing, Grazing);
} else { return 0; }
```

tan_into_housing Annual N flux as TAN into the house.

```
if( Val(tan_excretion, Excretion) > Val(tan_into_grazing, Grazing) ) {
Val(tan_excretion, Excretion)
- Val(tan_into_grazing, Grazing);
} else { return 0; }
```

nh3_nhousing Annual NH3 emission from small ruminants housing systems per animal place.

```
Out(tan_into_housing)
* Val(c_grazing, Housing::KGrazing)
* Tech(er_housing)
* Val(c_free_factor_housing, Housing::CFreeFactor) ;
```

n_outhousing Annual N flux out of the housing.

```
if( (Out(n_into_housing) - Out(nh3_nhousing)) > 0) {
Out(n_into_housing) - Out(nh3_nhousing);
} else { return 0; }
```

tan_outhousing Annual N flux as TAN out of the housing.

```
if( (Out(tan_into_housing) - Out(nh3_nhousing)) > 0) {
Out(tan_into_housing) - Out(nh3_nhousing);
} else { return 0; }
```

n_outhousing_liquid Annual N flux out of housing, slurry or liquid fraction of N flux.

```
0
```

tan_outhousing_liquid Annual N flux as TAN out of housing, slurry or liquid fraction of N flux.

```
0
```

n_outhousing_solid Annual N flux out of housing, manure fraction of N flux.

```
Out(n_outhousing);
```

tan_outhousing_solid Annual N flux as TAN out of housing, manure fraction of N flux.

```
Out(tan_outhousing);
```

1.36 Livestock::SmallRuminants::Housing::CFreeFactor

TODO

1.36.1 Inputs

free_correction_factor Factor to define free.

1.36.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a housing of the "
. "category small ruminants of " . In(free_correction_factor)
. "\%!\n",
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Stall der "
. "Kategorie Kleinwiederkäuer von " . In(free_correction_factor)
. "\% eingegeben!\n",
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions dans les "
."stabulations de la catégorie de petits ruminants de " . In(free_correction_factor)
. "\%.\n"
});
return 1 - In(free_correction_factor) / 100;
}
else {
return 1;
}
```

1.37 Livestock::SmallRuminants::Housing::KGrazing

This process calculates the correction factor for the non proportional change of the housing emission according to the grazing hours per day. Source is Phillips et al. (1998) and Gilhespy et al. (2006) empirical estimation was done by Thomas Kupper.

1.37.1 Outputs

c_grazing The correction factor for the reduction of the housing emission depending on the number of grazing hours per day and the grazing days per year.

```
my $k_grazing = Tech(k_grazing_a) * exp(Tech(k_grazing_b) * Val(grazing_hours, ..::Grazing));
# scale with ratio grazing_days per year
($k_grazing - 1) * Val(grazing_days, ..::Grazing) / 365 + 1;
```

1.38 Livestock::SmallRuminants::Grazing

This process calculates the annual NH3 emission from grazing goats, fattening sheep and milk-sheep based on the total N excreted on the pastures. The N excreted is calculated according to the time the animals spend on the pasture. It assumes that the excretions on the pasture are proportional to the time the animals are grazed.

1.38.1 References:

Bussink DW 1992. Ammonia volatilization from grassland receiving nitrogen fertilizer and rotationally grazed by dairy cattle. Fertilizer Research 33:257-265.

Bussink DW 1994. Relationship between ammonia volatilization and nitrogen fertilizer application rate, intake and excretion of herbage nitrogen by cattle on grazed swards. Fertilizer Research 38:111-121.

Jarvis SC, Hatch DJ, Lockyer DR 1989. Ammonia fluxes from grazed grassland: annual losses from cattle production systems and their relation to nitrogen inputs. Journal of Agricultural Science 113:99-108.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Ross CA, Jarvis SC 2001. Measurement of emission and deposition pattern of ammonia from urine in grass swards. Atmospheric environment 35:867-875.

1.38.2 Inputs

grazing_days Average grazing days per year.

grazing_hours Average grazing hours per day.

1.38.3 Outputs

grazing_hours Grazing hours per day.

```
In(grazing_hours);
```

grazing_days Grazing days per year.

```
In(grazing_days);
```

n_into_grazing Annual N excretion during grazing for small ruminants.

```
Val(n_excretion,Excretion) *
In(grazing_days)  / 365 *
In(grazing_hours) /  24;
```

tan_into_grazing Annual solable N (TAN) excretion during grazing for small ruminants.

```
Val(tan_excretion,Excretion) *
In(grazing_days)  / 365 *
In(grazing_hours) /  24;
```

ef_nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
Tech(er_small_ruminants_grazing);
```

nh3_ngrazing Annual NH3 emission from small ruminants from grazing.

```
Out(tan_into_grazing) * Out(ef_nh3_ngrazing);
```

n2_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2_small_ruminants_grazing);
```

no_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_no_small_ruminants_grazing);
```

n2o_ngrazing Annual total N2O emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2o_small_ruminants_grazing);
```

n_remain_grazing Annual N input on pastures.

```
Out(n_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

tan_remain_grazing Annual N input on pastures.

```
Out(tan_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
```

```
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

1.39 Livestock::SmallRuminants::Excretion

This process calculated the annual N excretion of small ruminats. The standard N excretion was taken from the official Swiss fertilizer guidelines. The Nsol content of the excreta is based on e.g. Peterson et al. (1998) or Burgos et al. (2005). These values were compiled on the basis of official feeding recommendations (Flisch et al. 2009) by H. Menzi.

1.39.1 References:

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.39.2 Inputs

animalcategory Animal category (Fattening sheep, milksheep and goats).

animals Number of other animals for the selected type in barn.

inp_n_excretion Annual standard N excretion for a dairy cow

1.39.3 Outputs

animals Number of small ruminants for the selected type in barn.

```
In(animals);
```

animalcategory Animal category

```
In(animalcategory);
```

n_excretion_animal Annual standard N excretion for specified other animal category according to Flisch et al. (2009).

```
my $cat = Out(animalcategory);
my $key = 'standard_N_excretion_' . $cat;
my $exc = $TE->{$key};
$exc = In(inp_n_excretion) unless lc In(inp_n_excretion) eq 'standard';
if ( ($exc < 0.25 * $TE->{$key}) or ($exc > 1.75 * $TE->{$key})) {
writeLog({
en => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
de => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
fr => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
});
}
return $exc;
```

n_excretion Annual N excreted by a specified number of small ruminants.

```
Out(n_excretion_animal) * Out(animals);
```

tan_excretion Annual soluble N excreted by a specified number of small ruminants.

```
$TE->{"share_Nsol_" . Out(animalcategory)} * Out(n_excretion);
```

1.40 Livestock::Equides

This process summarizes the annual emission of NH3 for equides from housing, yard and grazing. Further it summarizes the annual N flux from the housing to the storage for equides.

1.40.1 Inputs

ignore Just a work around for modules without input parameters.

1.40.2 Outputs

n2_nsolid Annual N2 emission from equides housing and yard (production).

```
( Val(n_into_housing, Equides::Housing) + Val(n_into_yard, Equides::Yard)
) *  Tech(er_n2_nsolid);
```

no_nsolid Annual NO emission from equides housing and yard (production).

```
( Val(n_into_housing, Equides::Housing) + Val(n_into_yard, Equides::Yard)
) *  Tech(er_no_nsolid);
```

n2o_nsolid Annual N2O emission from equides housing and yard (production).

```
( Val(n_into_housing, Equides::Housing) + Val(n_into_yard, Equides::Yard)
) *  Tech(er_n2o_nsolid);
```

1.41 Livestock::Equides::Housing

This process calculates the NH3 emission in equides housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emissions, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.41.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.41.2 Outputs

n_into_housing Annual N flux into the house.

```
if( Val(n_excretion, Excretion) > (Val(n_into_grazing, Grazing) + Val(n_into_yard, Yard)) ) {
Val(n_excretion, Excretion) -
Val(n_into_grazing, Grazing) -
Val(n_into_yard, Yard);
} else {
0;
}
```

tan_into_housing Annual N flux as TAN into the house.

```
if( Val(tan_excretion, Excretion) > (Val(tan_into_grazing, Grazing) + Val(tan_into_yard, Yard)) ) {
Val(tan_excretion, Excretion) -
Val(tan_into_grazing, Grazing) -
Val(tan_into_yard, Yard);
} else {
0;
}
```

nh3_nhousing Annual NH3 emission from equides housing systems.

```
my $c_housing = Val(c_grazing, Housing::KGrazing) *
Tech(er_housing) *
Val(c_free_factor_housing, Housing::CFreeFactor) ;
#FIXME: Check: Is it in any way even possible that nh3_nhousing > tan_into_housing???
$c_housing = 1 unless $c_housing < 1;
Out(tan_into_housing) * $c_housing;
```

n_outhousing Annual N flux out of the housing.

```
Out(n_into_housing) - Out(nh3_nhousing);
```

tan_outhousing Annual N flux as TAN out of the housing.

```
Out(tan_into_housing) - Out(nh3_nhousing);
```

n_outhousing_liquid Annual N flux out of housing, slurry or liquid fraction of N flux.

```
0
```

tan_outhousing_liquid Annual N flux as TAN out of housing, slurry or liquid fraction of N flux.

```
0
```

n_outhousing_solid Annual N flux out of housing, manure fraction of N flux.

```
Out(n_outhousing);
```

tan_outhousing_solid Annual N flux as TAN out of housing, manure fraction of N flux.

```
Out(tan_outhousing);
```

1.42 Livestock::Equides::Yard

1.42.1 References

Keck M 1997: Ammonia emission and odour thresholds of cattle houses with exercise yards. In: Voermans JAM and Monteny GJ (Eds): "Ammonia and odour emissions from animal production facilities", Proc. International Symposium, Vinkeloord, NL, 6-10 October 1997, 349-354. Misselbrook TH, Webb J, Chadwick DR, Ellis S, Pain BF 2001. Gaseous emissions from outdoor concrete yards used by livestock. Atmospheric Environment 35:5331-5338.

1.42.2 Inputs

yard_days Access to exercise yard in days per year.

yard_hours Access to exercise yard in hours per day.

floor_properties_exercise_yard_SHL Floor properties (solid_floor, unpaved_floor, paddock_or_pasture_used_as_exercise_yard).

floor_properties_exercise_yard_LU Floor properties (solid_floor, unpaved_floor).

free_correction_factor Factor to define free ?

1.42.3 Outputs

c_free_factor_yard Free correction factor of the Emission rate for the Yard.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({en=>"You have entered an additional emission mitigation measure for the exercise yard of "
. "the category horses and other equids of " . In(free_correction_factor)
. "\%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Laufhof der Kategorie "
. "Pferde und andere Equiden von "
. In(free_correction_factor)
. "\% eingegeben!\n",
fr=>"Vous avez introduit une mesure supplémentaire limitant les émissions provenant d'un "
. "parcours extérieur pour chevaux et autres équidés de " . In(free_correction_factor)
. "\%.\n" });
return 1 - In(free_correction_factor) / 100;
} else {
return 1;
}
```

c_floor_properties_exercise_yard_SHL Correction factor for the emission due to the use of the floor properties in housing systems.

```
given ( In(floor_properties_exercise_yard_SHL) ) {
when 'unpaved_floor' {
1 - Tech(red_floor_properties_unpaved_floor);
}
when 'solid_floor' {
1 - Tech(red_floor_properties_solid_floor);
}
when 'paddock_or_pasture_used_as_exercise_yard' {
1 - Tech(red_floor_properties_paddock_or_pasture_used_as_exercise_yard);
}
default {
1;
}
}
```

c_floor_properties_exercise_yard_LU Correction factor for the emission due to the use of the floor properties in housing systems.

```
return 1 unless defined In(floor_properties_exercise_yard_LU);
given ( In(floor_properties_exercise_yard_LU) ) {
when 'unpaved_floor' {
1 - Tech(red_floor_properties_unpaved_floor);
}
when 'solid_floor' {
1 - Tech(red_floor_properties_solid_floor);
}
default {
1;
}
}
```

n_into_yard Annual N excretion on yard for a defined animal category.

```
Val(n_excretion, Excretion) *
In(yard_days)  / 365 *
In(yard_hours) / 24;
```

tan_into_yard Annual soluble N excretion on yard for a defined animal category.

```
Val(tan_excretion, Excretion) *
In(yard_days)  / 365 *
In(yard_hours) / 24;
```

ef_nh3_nyard NH3 emission factor for dairy cow yard.

```
Tech(er_yard) *
Out(c_floor_properties_exercise_yard_SHL) *
Out(c_floor_properties_exercise_yard_LU) *
Out(c_free_factor_yard);
```

nh3_nyard Annual NH3 emission from yard.

```
Out(tan_into_yard) * Out(ef_nh3_nyard);
```

n_outyard_liquid Annual N flux from liquid part out of yard.

```
0;
```

tan_outyard_liquid Annual N flux as TAN from liquid part out of yard into storage.

```
0;
```

n_outyard_solid Annual N flux from solid part out of yard.

```
Out(n_into_yard) - Out(nh3_nyard);
```

tan_outyard_solid Annual N flux as TAN from solid part out of yard into storage.

```
Out(tan_into_yard) - Out(nh3_nyard);
```

1.43 Livestock::Equides::Housing::CFreeFactor

TODO

1.43.1 Inputs

free_correction_factor Factor to define free.

1.43.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en=>"You have entered an additional emission mitigation measure for a housing of the "
. "category horses and other equids of " . In(free_correction_factor)
. "\%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Stall der "
. "Kategorie Pferde und andere Equiden von " . In(free_correction_factor)
. "\% eingegeben!\n",
fr=>"Vous avez introduit une mesure supplémentaire limitant les émissions dans les "
."stabulations pour chevaux et autres équidés " . In(free_correction_factor)
. "\%.\n"
});
return 1 - In(free_correction_factor) / 100;
}
else {
return 1;
}
```

1.44 Livestock::Equides::Housing::KGrazing

This process calculates the correction factor for the non proportional change of the housing emission according to the grazing hours per day. Source is Phillips et al. (1998) and Gilhespy et al. (2006) empirical estimation was done by Thomas Kupper.

1.44.1 Outputs

c_grazing The correction factor for the reduction of the housing emission depending on the number of grazing hours per day and the grazing days per year.

```
my $k_grazing = Tech(k_grazing_a) * exp(Tech(k_grazing_b) * Val(grazing_hours, ..::Grazing));
# scale with ratio grazing_days per year
($k_grazing - 1) * Val(grazing_days, ..::Grazing) / 365 + 1;
```

1.45 Livestock::Equides::Grazing

This process calculates the annual NH3 emission from grazing of equides (horses, mules, asses) based on the total N excreted on the pastures. The N excreted is calculated according to the time the animals spend on the pasture. It assumes that the excretions on the pasture are proportional to the time the animals are grazed.

1.45.1 References:

Bussink DW 1992. Ammonia volatilization from grassland receiving nitrogen fertilizer and rotationally grazed by dairy cattle. Fertilizer Research 33:257-265.

Bussink DW 1994. Relationship between ammonia volatilization and nitrogen fertilizer application rate, intake and excretion of herbage nitrogen by cattle on grazed swards. Fertilizer Research 38:111-121.

Jarvis SC, Hatch DJ, Lockyer DR 1989. Ammonia fluxes from grazed grassland: annual losses from cattle production systems and their relation to nitrogen inputs. Journal of Agricultural Science 113:99-108.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Ross CA, Jarvis SC 2001. Measurement of emission and deposition pattern of ammonia from urine in grass swards. Atmospheric environment 35:867-875.

1.45.2 Inputs

grazing_days Average grazing days per year.

grazing_hours Average grazing hours per day.

1.45.3 Outputs

grazing_hours Grazing hours per day.

```
In(grazing_hours);
```

grazing_days Grazing days per year.

```
In(grazing_days);
```

n_into_grazing Annual N excretion during grazing for equides.

```
Val(n_excretion, Excretion) *
In(grazing_days)  / 365 *
In(grazing_hours) /  24;
```

tan_into_grazing Annual solable N (TAN) excretion during grazing for equides.

```
Val(tan_excretion,Excretion) *
In(grazing_days)  / 365 *
In(grazing_hours) /  24;
```

ef_nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
Tech(er_equides_grazing);
```

nh3_ngrazing Annual NH3 emission from equides from grazing.

```
Out(tan_into_grazing) * Out(ef_nh3_ngrazing);
```

n2_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2_equides_grazing);
```

no_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_no_equides_grazing);
```

n2o_ngrazing Annual total N2O emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2o_equides_grazing);
```

n_remain_grazing Annual N input on pastures.

```
Out(n_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

tan_remain_grazing Annual N input on pastures.

```
Out(tan_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

1.46 Livestock::Equides::Excretion

This process calculated the annual N excretion of the animal categories listed above. The standard N excretion was taken from the official Swiss fertilizer guidelines. The Nsol content of the excreta is based on e.g. Peterson et al. (1998) or Burgos et al. (2005). These values were compiled on the basis of official feeding recommendations (Flisch et al. 2009) by H. Menzi.

1.46.1 References:

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.46.2 Inputs

animalcategory Animal category (horses younger than 3 years, horses older than 3 years, mules, ponies and asses).

animals Number of other animals for the selected type in barn.

inp_n_excretion Annual standard N excretion for animal category

1.46.3 Outputs

animals Number of other animals for the selected type in barn.

```
In(animals);
```

animalcategory Animal category

```
In(animalcategory);
```

n_excretion_animal Annual standard N excretion for specified other animal category according to Flisch et al. (2009).

```
my $cat = In(animalcategory);
my $key = 'standard_N_excretion_' . $cat;
my $exc = $TE->{$key};
$exc = In(inp_n_excretion) unless lc In(inp_n_excretion) eq 'standard';
if ( ($exc < 0.25 * $TE->{$key}) or ($exc > 1.75 * $TE->{$key})) {
writeLog({
en => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
de => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
fr => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
});
}
return $exc;
```

n_excretion Annual N excreted by a specified number of animals.

```
Out(n_excretion_animal) * Out(animals);
```

tan_excretion Annual soluble N excreted by a specified number of animals.

```
$TE->{"share_Nsol_" . Out(animalcategory)} * Out(n_excretion);
```

1.47 Livestock::Poultry

This process summarizes the annual emission of NH3, the annual N excretion, and the annual N flux for different poultry categories from housing and outdoor.

1.47.1 Inputs

ignore Just a work around for modules without input parameters.

1.47.2 Outputs

n_excretion_layers_growers_other_poultry Annual N excreted by poultry.

```
if (Val(animalcategory, Poultry::Excretion) eq 'layers'
|| Val(animalcategory, Poultry::Excretion) eq 'growers'
|| Val(animalcategory, Poultry::Excretion) eq 'other_poultry'){
Val(n_excretion, Poultry::Excretion);
}else{
return(0);
};
```

n_excretion_turkeys_broilers Annual N excreted by poultry.

```
if( Val(animalcategory, Poultry::Excretion ) eq 'turkeys'
|| Val(animalcategory, Poultry::Excretion ) eq 'broilers'){
Val(n_excretion, Poultry::Excretion);
}else{
return(0);
};
```

n_excretion Annual N excreted by poultry.

```
Val(n_excretion, Poultry::Excretion);
```

n2_npoultry Annual N2 emission from poultry production (housing and outdoor).

```
Val(n_into_housing, Poultry::Housing) * Tech(er_n2)
```

no_npoultry Annual N2 emission from poultry production housing. (outdoor not included)

```
Val(n_into_housing, Poultry::Housing) * Tech(er_no)
```

n2o_npoultry Annual N2O emission from poultry production housing. (outdorr not included)

```
Val(n_into_housing, Poultry::Housing) * Tech(er_n2o)
```

n_from_poultry_layers_growers_other_poultry Annual N flux from poultry housing.

```
if( Val(animalcategory, Poultry::Excretion) eq 'layers'
|| Val(animalcategory, Poultry::Excretion) eq 'growers'
||  Val(animalcategory, Poultry::Excretion) eq 'other_poultry'){
Val(n_outhousing_solid, Poultry::Housing);
}else{
return(0);
};
```

n_from_poultry_turkeys_broilers Annual N flux from poultry housing.

```
if( Val(animalcategory,Poultry::Excretion) eq 'turkeys'
|| Val(animalcategory,Poultry::Excretion) eq 'broilers') {
Val(n_outhousing_solid, Poultry::Housing);
}else{
return(0);
};
```

n_from_poultry Annual N flux from poultry housing.

```
Val(n_outhousing_solid, Poultry::Housing)
```

tan_from_poultry_layers_growers_other_poultry Annual N flux from poultry housing.

```
if( Val(animalcategory, Poultry::Excretion) eq 'layers'
|| Val(animalcategory, Poultry::Excretion) eq 'growers'
||  Val(animalcategory, Poultry::Excretion) eq 'other_poultry'){
Val(tan_outhousing_solid, Poultry::Housing)
}else{
return(0);
};
```

tan_from_poultry_turkeys_broilers Annual N flux from poultry housing.

```
if( Val(animalcategory, Poultry::Excretion) eq 'turkeys'
|| Val(animalcategory, Poultry::Excretion) eq 'broilers'){
Val(tan_outhousing_solid, Poultry::Housing)
}else{
return(0);
};
```

tan_solid Annual N flux from poultry housing.

```
Val(tan_outhousing_solid, Poultry::Housing)
```

tan_liquid Annual N flux from poultry housing (only for detail report).

```
0
```

tan Annual N flux from poultry housing.

```
Out(tan_solid)
```

nh3_npoultry_layers_growers_other_poultry Annual NH3 emission from poultry production (housing and outdoor).

```
if( Val(animalcategory, Poultry::Excretion ) eq 'layers'
|| Val(animalcategory, Poultry::Excretion ) eq 'growers'
||  Val(animalcategory, Poultry::Excretion ) eq 'other_poultry'){
Val(nh3_nhousing, Poultry::Housing) +
Val(nh3_free_range, Poultry::Outdoor);
}else{
return(0);
};
```

n2_npoultry_layers_growers_other_poultry Annual N2 emission from poultry production if layer , growers or other poultry

```
if( Val(animalcategory, Poultry::Excretion ) eq 'layers'
|| Val(animalcategory, Poultry::Excretion ) eq 'growers'
|| Val(animalcategory, Poultry::Excretion ) eq 'other_poultry'){
Out(n2_npoultry);
}else{
return(0);
};
```

no_npoultry_layers_growers_other_poultry Annual NO emission from poultry production if layer , growers or other poultry

```
if( Val(animalcategory, Poultry::Excretion ) eq 'layers'
|| Val(animalcategory, Poultry::Excretion ) eq 'growers'
||  Val(animalcategory, Poultry::Excretion ) eq 'other_poultry'){
Out(no_npoultry);
}else{
return(0);
};
```

n2o_npoultry_layers_growers_other_poultry Annual N2O emission from poultry production if layer , growers or other poultry

```
if( Val(animalcategory, Poultry::Excretion ) eq 'layers'
|| Val(animalcategory, Poultry::Excretion ) eq 'growers'
||  Val(animalcategory, Poultry::Excretion ) eq 'other_poultry'){
Out(n2o_npoultry);
}else{
return(0);
};
```

nh3_npoultry_turkeys_broilers Annual NH3 emission from poultry production (housing and outdoor).

```
if( Val(animalcategory, Poultry::Excretion ) eq 'turkeys'
|| Val(animalcategory, Poultry::Excretion ) eq 'broilers'){
Val(nh3_nhousing, Poultry::Housing) +
```

```
Val(nh3_free_range, Poultry::Outdoor);
}else{
return(0);
};
```

n2_npoultry_turkeys_broilers Annual N2 emission from poultry production if turkey or broiler

```
if( Val(animalcategory, Poultry::Excretion ) eq 'turkeys'
|| Val(animalcategory, Poultry::Excretion ) eq 'broilers'){
Out(n2_npoultry)
}else{
return(0);
};
```

no_npoultry_turkeys_broilers Annual NO emission from poultry production if turkey or broiler

```
if( Val(animalcategory, Poultry::Excretion ) eq 'turkeys'
|| Val(animalcategory, Poultry::Excretion ) eq 'broilers'){
Out(no_npoultry)
}else{
return(0);
};
```

n2o_npoultry_turkeys_broilers Annual N2O emission from poultry production if turkey or broiler

```
if( Val(animalcategory, Poultry::Excretion ) eq 'turkeys'
|| Val(animalcategory, Poultry::Excretion ) eq 'broilers'){
Out(n2o_npoultry)
}else{
return(0);
};
```

nh3_npoultry Annual NH3 emission from poultry production (housing and outdoor).

```
Val(nh3_nhousing, Poultry::Housing) +
Val(nh3_free_range, Poultry::Outdoor);

#+animalcategory
# print = 7
# ++units
#   en = -
# ++description
#   Animal category
# ++formula
#   Val(animalcategory, Poultry::Excretion);
```

1.48 Livestock::Poultry::Housing

This process calculates the NH3 emission in poultry housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emissions, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.48.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.48.2 Outputs

n_into_housing Annual N flux into the housing.

```
Val(n_excretion, Excretion)
- Val(n_into_outdoor, Outdoor);
```

tan_into_housing Annual N flux into the housing.

```
Val(tan_excretion, Excretion)
- Val(tan_into_outdoor, Outdoor);
```

nh3_nhousing Annual NH3 emission from poultry housing systems per animal place.

```
(Val(n_excretion, Excretion)
* Val(er_housing, Housing::Type)
* Val(c_area, Housing::KArea)
* Val(c_manure_removal_interval, Housing::Type)
* Val(c_drinking_system, Housing::Type)
* (1 - Val(c_air_scrubber, Housing::AirScrubber)))
* Val(c_free_factor_housing, Housing::CFreeFactor);
```

n_housing_filter Annual N of NH3 emission remaining in air scrubber from poultry housing systems.

```
Val(n_excretion, Excretion)
* Val(er_housing, Housing::Type)
* Val(c_manure_removal_interval, Housing::Type)
* Val(c_drinking_system, Housing::Type)
* Val(c_air_scrubber, Housing::AirScrubber);
```

n_outhousing_solid Annual N flux out of the housing including N remained in biotrickling filter.

```
if(Val(air_scrubber, Housing::AirScrubber) eq 'acid'){
Out(n_into_housing) - Out(nh3_nhousing) - Out(n_housing_filter)
}else{
Out(n_into_housing) - Out(nh3_nhousing)
};
```

tan_outhousing_solid Annual N flux out of the housing including N remained in biotrickling filter.

```
if(Val(air_scrubber, Housing::AirScrubber) eq 'acid'){
Out(tan_into_housing) - Out(nh3_nhousing) - Out(n_housing_filter)
}else{
Out(tan_into_housing) - Out(nh3_nhousing)
};
```

1.49 Livestock::Poultry::Housing::CFreeFactor

This process selects the emission rate for the specific housing types for poultry and the correction factors for the drinking system, and for the manure removal interval.

1.49.1 Inputs

free_correction_factor Factor to define free.

1.49.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a poultry housing of "
. In(free_correction_factor) . "\%!\n",
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Geflügelstall von "
. In(free_correction_factor) . "\% eingegeben!\n",
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions provenant d'une "
. "stabulation pour volaille de " . In(free_correction_factor) . "\%.\n"
});
return 1 - In(free_correction_factor) / 100;
}
else {
return 1;
}
```

1.50 Livestock::Poultry::Housing::AirScrubber

This submodul calculates the annual NH3 reduction due to an exhaust air scrubber in housing systems according to the UNECE guideline 2007.

1.50.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.50.2 Inputs

air_scrubber Exhaust air scrubber: none, acid, biotrickling_filter.

1.50.3 Outputs

air_scrubber air exhaust scrubber in housing systems.

```
In(air_scrubber);
```

c_air_scrubber Reduction factor for the emission due to the use of an air exhaust scrubber in housing systems.

```
given ( In(air_scrubber) ) {
when 'acid' {
return Tech(red_acid_air_scrubber);
}
when 'biotrickling' {
return Tech(red_biotrickling_filter_air_scrubber);
}
when 'none' {
return 0;
}
default {
writeLog({
en => "Invalid 'exhaust air scrubber' entered!",
de => "Ungültige 'Abluftreinigung' eingegeben!",
fr => "Entrée non valable pour 'Traitement des effluents gazeux'!"
});
return 0;
}
}
```

1.51 Livestock::Poultry::Housing::Type

This process selects the emission rate for the specific housing types for poultry and the correction factors for the drinking system, and for the manure removal interval.

1.51.1 References

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13. Reidy B, Webb J, Misselbrook TH, Menzi H, Luesink HH, Hutchings NJ, Eurich-Menden B, Döhler H, Dämmgen U 2009. Comparison of models used for national agricultural ammonia emission inventories in Europe: litter-based manure systems. Atmospheric Environment 40, 1632-1640.

1.51.2 Inputs

housing_type Type of housing.

manure_removal_interval Manure removal interval by manure belt.

drinking_system Type of drinking system.

1.51.3 Outputs

er_housing Emission rate for the poultry housing type.

```
if ( Val(animalcategory, ..::Excretion) eq "layers" or
Val(animalcategory, ..::Excretion) eq "growers" ) {
if (In(housing_type) eq "manure_belt_without_manure_belt_drying_system"){
return Tech(er_housing_layers_growers_manure_belt_without_manure_belt_drying_system);
}
elsif (In(housing_type) eq "manure_belt_with_manure_belt_drying_system"){
return Tech(er_housing_layers_growers_manure_belt_with_manure_belt_drying_system);
}
elsif (In(housing_type) eq "deep_pit"){
return Tech(er_housing_layers_growers_deep_pit);
}
elsif(In(housing_type) eq "deep_litter"){
return Tech(er_housing_layers_growers_deep_litter);
}
} else {
if ( In(housing_type) eq "manure_belt_without_manure_belt_drying_system" ) {
writeLog({
en => "The category manure belt without manure belt drying system is not valid, please select the housing sys
de => "Für Mastpoulets ist Kotbandentmistung als Aufstallung nicht vorgesehen. Wählen Sie stattdessen Bodenha
. "(für die Berechnung wird Bodenhaltung gesetzt).",
fr => "Pour des poulets à l‘engrais, il n’est pas prévu de stabulation avec "
. "tapis d’évacuation du fumier. Choisissez plutôt ‘Litière profonde’."
});
}
elsif ( In(housing_type) eq "manure_belt_with_manure_belt_drying_system" ) {
writeLog({
en => "Manure belt with manure belt drying system not valid, please select deep litter as housing type for br
de => "Für Mastpoulets ist Kotbandentmistung als Aufstallung nicht vorgesehen. Wählen Sie stattdessen Bodenha
. "(für die Berechnung wird Bodenhaltung gesetzt).",
fr => "Pour des poulets à l‘engrais, il n’est pas prévu de stabulation avec "
. "tapis d’évacuation du fumier. Choisissez plutôt ‘Litière profonde’."
});
}
elsif ( In(housing_type) eq "deep_pit" ) {
```

```
writeLog({
en => "Deep pit not valid, please select deep litter as housing type for broilers.",
de => "Für Mastpoulets ist Kotgrube als Aufstallung nicht vorgesehen. Wählen Sie stattdessen Bodenhaltung. "
. "(für die Berechnung wird Bodenhaltung gesetzt).",
fr => "Pour des poulets à l‘engrais, il n’est pas prévu de stabulation avec "
. "fosse à déjections. Choisissez plutôt ‘Litière profonde’."
});
}
return Tech(er_housing_other_deep_litter);
}


# +c_manure_removal_interval
#   print = 10
#   ++units
#     en = -
#   ++description
#     Emission rate for the poultry housing type.
#   ++formula
#     if ( In(housing_type) eq "manure_belt_without_manure_belt_drying_system" or
#          In(housing_type) eq "manure_belt_with_manure_belt_drying_system" ) {
#       switch ( In(manure_removal_interval) ) {
#         case "less_than_twice_a_month" { return Tech(c_manure_removal_interval_less_than_twice_a_month) }
#         case "twice_a_month" { return Tech(c_manure_removal_interval_twice_a_month) }
#         case "3_to_4_times_a_month" { return Tech(c_manure_removal_interval_3_to_4_times_a_month) }
#         case "more_than_4_times_a_month" { return Tech(c_manure_removal_interval_more_than_4_times_a_month)
#         case "once_a_day" { return Tech(c_manure_removal_interval_once_a_day) }
#         else {
#           writeLog({
#             en => "The category ’No manure belt’ is not applicable for housing systems with a manure belt!"
#             de => "Kategorie ’keine Kotbandentmistung’, ist nicht erlaubt bei Aufstallung, Kotbandentmistur
#             fr => "La catégorie ‘Pas de tapis d’évacuation’ n’est pas valable pour une "
#                   . "stabulation avec tapis d’évacuation du fumier!",
#           });
#           return Tech(c_manure_removal_interval_twice_a_month);
#         }
#       }
#     }
#     else { # Housing Type deep pit or deep litter
#                   if( not (In(manure_removal_interval) eq "no_manure_belt") ){
#                   writeLog({en => "Please enter under manure removal interval ’No manure belt’ in combinati
#                         de => "Bitte wählen Sie unter Entmistungsintervall ’keine Kotbandentmistung’
#                        fr => "Sous ‘Fréquence d’évacuation du fumier’, veuillez choisir la catégorie "
#                               . "‘Pas de tapis d’évacuation’ en combinaison avec la stabulation
#                            . "profonde’ ou ‘Fosse à déjections’ !",
#                          });
#           }
#                   return 1; # default no correctios
#     }
```

c_manure_removal_interval Emission rate for the poultry housing type.

```
if ( (In(housing_type) eq "manure_belt_without_manure_belt_drying_system")
or (In(housing_type) eq "manure_belt_with_manure_belt_drying_system")) {
if (In(manure_removal_interval) eq "less_than_twice_a_month"){
return Tech(c_manure_removal_interval_less_than_twice_a_month);
} elsif(In(manure_removal_interval) eq "twice_a_month"){
return Tech(c_manure_removal_interval_twice_a_month);
} elsif(In(manure_removal_interval) eq "3_to_4_times_a_month"){
return Tech(c_manure_removal_interval_3_to_4_times_a_month);
} elsif(In(manure_removal_interval) eq "more_than_4_times_a_month"){
return Tech(c_manure_removal_interval_more_than_4_times_a_month);
} elsif(In(manure_removal_interval) eq "once_a_day"){
return Tech(c_manure_removal_interval_once_a_day);
} elsif(In(manure_removal_interval) eq "no_manure_belt") {
writeLog({en => "The category ’No manure belt’ is not applicable for housing systems with a manure belt!",
de => "Kategorie ’keine Kotbandentmistung’, ist nicht erlaubt bei Aufstallung, Kotbandentmistung!",
fr => "La catégorie ‘Pas de tapis d’évacuation’ n’est pas valable pour une "
```

```
. "stabulation avec tapis d’évacuation du fumier!",
});
return Tech(c_manure_removal_interval_twice_a_month);  # default no correctios
}
}
else{                                            # Housing Type deep pit or deep litter
if( not (In(manure_removal_interval) eq "no_manure_belt") ){
writeLog({en => "Please enter under manure removal interval ’No manure belt’ in combination with the housing
de => "Bitte wählen Sie unter Entmistungsintervall ’keine Kotbandentmistung’ in Kombination mit der Aufstallu
fr => "Sous ‘Fréquence d’évacuation du fumier’, veuillez choisir la catégorie "
. "‘Pas de tapis d’évacuation’ en combinaison avec la stabulation ‘Litière "
. "profonde’ ou ‘Fosse à déjections’ !",
});
}
return 1; # default no correctios
}
```

c_drinking_system Correction factor for poultry drinking station.

```
if (In(drinking_system) eq "drinking_nipples"){
return Tech(c_drinking_nipples);
} elsif(In(drinking_system) eq "bell_drinkers"){
return Tech(c_bell_drinkers);
}else {
writeLog({en => "Invalid ’drinking_system’ entered!",
de => "Ungültiges Tränkesystem eingegeben!",
en => "Ce type d’abreuvoir n’est pas valable !",
});
return 1.0;
}
```

housing_type Housing type (needed in other modules).

```
In(housing_type);
```

drinking_system Drinking system.

```
In(drinking_system);
```

1.52 Livestock::Poultry::Housing::KArea

It is assumed that additional areas in stable, as example if the stabel is built for more animal, are fully used. So the emissions increase 5
the double of the regularized minimal area.

TODO (All): Provide Articles which confirm/justify the assumption. As decided on workshop from 2.11.2007.

1.52.1 Inputs

dimensioning_barn Number of available animal places.

1.52.2 Outputs

dimensioning_barn Number of anymal places (needed in other modules).

```
In(dimensioning_barn);
```

area_increase Factor on what barn size does increase the regularized minimal, limited to 0.5

```
if ( (Val(animals, ..::Excretion) < In(dimensioning_barn) ) and (Val(animals, ..::Excretion) != 0) ) {
if ( In(dimensioning_barn) >= (Val(animals, ..::Excretion) * 1.5) ) {
0.5;
}
else {
In(dimensioning_barn) / Val(animals, ..::Excretion) - 1;
}
}
else {
0.0;
}
```

c_area Correction factor for area per animal.

```
1 + (Out(area_increase) * Tech(k_area));
```

1.53 Livestock::Poultry::Outdoor

This process calculates the annual NH3 emission of free range poultry depending on the free range N excretion and the emission rate. The annual N excretion calculation is based on the grazing hours per day per year and the free range hours per day per year. The annual remaining N from free range poultry is calculated as the annual N excretion minus the annual NH3 emission.

1.53.1 References:

Menzi H, Shariatmadari H, Meierhans D, Wiedmer H 1997: Nähr- und Schadstoffbelastung von Geflügelausläufen. Agrarforschung 4: 361-364.

1.53.2 Inputs

free_range Average free range hours per day.

1.53.3 Outputs

free_range_days Average free range days per year.

```
if( lc In(free_range) eq 'yes' ){
#       my $key = "free_range_days_" . Val(animalcategory, Excretion);
#       return Tech($key);
return $TE->{'free_range_days_'.Val(animalcategory,Excretion)};
}else{
return 0;
}
```

free_range_hours Average free range hours per day.

```
if( lc In(free_range) eq 'yes' ){
#       my $key = "free_range_hours_" .  Val(animalcategory, Excretion);
#       return Tech($key);
return $TE->{'free_range_hours_'.Val(animalcategory,Excretion)};
}else{
return 0;
}
```

n_into_outdoor Annual N excretion free_range (grazing).

```
Val(n_excretion, Excretion)   *
( (Out(free_range_days)/365) * (Out(free_range_hours)/24) );
```

tan_into_outdoor Annual N excretion free_range (grazing).

```
Val(tan_excretion, Excretion) *
( (Out(free_range_days)/365) * (Out(free_range_hours)/24)) ;
```

nh3_free_range Annual free_range NH3 emission from poultry (grazing).

```
Out(tan_into_outdoor) * Tech(er_free_range);
```

n2_free_range Annual free_range N2 emission from poultry (grazing).

```
Out(n_into_outdoor) * Tech(er_n2_free_range);
```

no_free_range Annual free_range NO emission from poultry (grazing).

```
Out(n_into_outdoor) * Tech(er_no_free_range);
```

n2o_free_range Annual free_range N2O emission from poultry (grazing).

```
Out(n_into_outdoor) * Tech(er_n2o_free_range);
```

n_remain_outdoor Annual N remaining free_range (on pasture, etc.).

```
Out(n_into_outdoor) -
Out(nh3_free_range) -
Out(no_free_range) -
Out(n2o_free_range);
```

tan_remain_outdoor Annual N remaining free_range (on pasture, etc.).

```
Out(tan_into_outdoor) -
Out(nh3_free_range) -
Out(no_free_range) -
Out(n2o_free_range);
```

1.54 Livestock::Poultry::Excretion

This process calculates the annual N excretion of the different poultry categories. The standard N excretion was taken from the official Swiss fertilizer guidelines. The Nsol content of the excreta is based on e.g. Peterson et al. (1998) or Burgos et al. (2005). These values were compiled on the basis of official feeding recommendations (Flisch et al. (2009)) by H. Menzi.

1.54.1 References:

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.54.2 Inputs

animalcategory Poultry category (layers, growers, broilers, turkeys, and other poultry).

animals Number of poultry animals for the selected type in barn.

inp_n_excretion Annual standard N excretion for poultry

1.54.3 Outputs

animals Number of poultry animals for the selected type in barn.

```
In(animals);
```

animalcategory Poultry category (layers, growers, broilers, turkeys, and other poultry).

```
In(animalcategory);
```

n_excretion_animal Annual standard N excretion for specified poultry category according to Flisch et al. (2009).

```
my $cat = Out(animalcategory);
my $key = 'standard_N_excretion_' . $cat;
my $exc = $TE->{$key};
$exc = In(inp_n_excretion) unless lc In(inp_n_excretion) eq 'standard';
if (($exc < 0.25 * $TE->{$key}) or ($exc > 1.75 * $TE->{$key})) {
writeLog({
en => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
de => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
fr => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
});
}
return $exc;
```

n_excretion Annual N excreted by a specified number of animals.

```
Out(n_excretion_animal) * Out(animals);
```

tan_excretion Annual soluble N excreted by an animalgroup of selected poultry category.

```
$TE->{'share_Nsol_'.Out(animalcategory)} * Out(n_excretion);
```

1.55 Livestock::FatteningPigs

This process summarizes the annual emission of NH3 for fattening pigs from housing and grazing. And it summarizes the annual N flux from housing to the storage for fattening pigs.

1.55.1 Outputs

er_n2_nsolid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
given ( Val(housing_type, FatteningPigs::Housing::Type) ) {
return Tech(er_n2_solid_Slurry) when 'Slurry_Conventional';
return Tech(er_n2_solid_Slurry) when 'Slurry_Label';
return Tech(er_n2_solid_Solid)  when 'Deep_Litter';
return Tech(er_n2_solid_Solid)  when 'Outdoor';
default { return 0; }
};
```

er_no_nsolid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
given ( Val(housing_type, FatteningPigs::Housing::Type) ) {
return Tech(er_no_solid_Slurry) when 'Slurry_Conventional';
return Tech(er_no_solid_Slurry) when 'Slurry_Label';
return Tech(er_no_solid_Solid)  when 'Deep_Litter';
return Tech(er_no_solid_Solid)  when 'Outdoor';
default { return 0; }
};
```

er_n2o_nsolid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
given ( Val(housing_type, FatteningPigs::Housing::Type) ) {
return Tech(er_n2o_solid_Slurry)  when 'Slurry_Conventional';
return Tech(er_n2o_solid_Slurry)  when 'Slurry_Label';
return Tech(er_n2o_solid_Solid)   when 'Deep_Litter';
return Tech(er_n2o_solid_Solid)   when 'Outdoor';
default { return 0; }
};
```

er_n2_nliquid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
given ( Val(housing_type, FatteningPigs::Housing::Type) ) {
return Tech(er_n2_liquid_Slurry)  when 'Slurry_Conventional';
return Tech(er_n2_liquid_Slurry)  when 'Slurry_Label';
return Tech(er_n2_liquid_Solid)   when 'Deep_Litter';
return Tech(er_n2_liquid_Solid)   when 'Outdoor';
default { return 0; }
};
```

er_no_nliquid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
given ( Val(housing_type, FatteningPigs::Housing::Type) ) {
return Tech(er_no_liquid_Slurry)  when 'Slurry_Conventional';
return Tech(er_no_liquid_Slurry)  when 'Slurry_Label';
return Tech(er_no_liquid_Solid)   when 'Deep_Litter';
return Tech(er_no_liquid_Solid)   when 'Outdoor';
default { return 0; }
};
```

er_n2o_nliquid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
given ( Val(housing_type, FatteningPigs::Housing::Type) ) {
return Tech(er_n2o_liquid_Slurry) when 'Slurry_Conventional';
return Tech(er_n2o_liquid_Slurry) when 'Slurry_Label';
return Tech(er_n2o_liquid_Solid)  when 'Deep_Litter';
return Tech(er_n2o_liquid_Solid)  when 'Outdoor';
default { return 0; }
};
```

n2_nsolid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
(
Val(n_into_housing, FatteningPigs::Housing) *
(1 - Val(share_liquid, FatteningPigs::Housing::Type))
) * Out(er_n2_nsolid);
```

no_nsolid_fattening_pig Annual NO emissions from fattening pigs housing, yard and grazing (production).

```
(
Val(n_into_housing, FatteningPigs::Housing) *
(1 - Val(share_liquid, FatteningPigs::Housing::Type))
) * Out(er_no_nsolid);
```

n2o_nsolid Annual N2O emissions from fattening pigs housing, yard and grazing (production).

```
(
Val(n_into_housing, FatteningPigs::Housing) *
(1 - Val(share_liquid, FatteningPigs::Housing::Type))
) * Out(er_n2o_nsolid);
```

n2_nliquid Annual N2 emissions from fattening pigs housing, yard and grazing (production).

```
( Val(n_into_housing, FatteningPigs::Housing) * Val(share_liquid, FatteningPigs::Housing::Type)
) * Out(er_n2_nliquid);
```

no_nliquid_fattening_pig Annual NO emissions from fattening pigs housing, yard and grazing (production).

```
( Val(n_into_housing, FatteningPigs::Housing) * Val(share_liquid, FatteningPigs::Housing::Type)
) * Out(er_no_nliquid);
```

n2o_nliquid Annual N2O emissions from fattening pigs housing, yard and grazing (production).

```
( Val(n_into_housing, FatteningPigs::Housing) * Val(share_liquid, FatteningPigs::Housing::Type)
) *  Out(er_n2o_nliquid);
```

1.56 Livestock::FatteningPigs::Housing

This process calculates the NH3 emission in fattening pig housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emissions, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.56.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.56.2 Outputs

n_into_housing Annual N flux into the house.

```
Val(n_excretion, Excretion) - Val(n_into_grazing, Grazing);
```

tan_into_housing Annual N flux as TAN into the house.

```
Val(tan_excretion, Excretion) - Val(tan_into_grazing, Grazing);
```

tan_into_housing_indoor Annual N flux as TAN into the house.

```
Out(tan_into_housing) * Val(share_indoor, Housing::Type);
```

nh3_nhousing Annual NH3 emission from fattening pig housing systems.

```
Out(tan_into_housing) *
Val(er_housing, Housing::Type) *
Val(c_area, Housing::KArea) *
Val(c_free_factor_housing, Housing::CFreeFactor) *
(
# indoor part + mitigation
Val(share_indoor, Housing::Type) *
(1 - Val(red_air_scrubber, Housing::AirScrubber)) *
(1 - Val(red_housing_floor, Housing::MitigationOptions)) *
(1 - Val(red_housing_air, Housing::MitigationOptions)) +
# outdoor part + mitigation
(1 - Val(share_indoor, Housing::Type))
);
```

n_air_scrubber Annual N of NH3 emission remaining in air scrubber from pig housing systems.

```
Val(red_air_scrubber, Housing::AirScrubber) *
# multiplied with indoor loss without air scrubber mitigation
Val(tan_excretion,Excretion) *
Val(er_housing, Housing::Type) *
Val(c_area, Housing::KArea) *
Val(c_free_factor_housing, Housing::CFreeFactor) *
Val(share_indoor, Housing::Type) *
(1 - Val(red_housing_floor, Housing::MitigationOptions)) *
(1 - Val(red_housing_air, Housing::MitigationOptions));
```

n_outhousing Annual N flux out of the housing including N remained in biotrickling filter, (without remains in acid filter).

```
if ( Val(housing_type,Housing::Type) eq 'Outdoor' ) {
0;
} else {
if ( Val(air_scrubber,Housing::AirScrubber) eq 'biotrickling' ) {
# n in biotrickling vanishes
Out(n_into_housing) - Out(nh3_nhousing) - Out(n_air_scrubber);
} else {
# n in acid scrubber adds 100% to flux into storage
Out(n_into_housing) - Out(nh3_nhousing);
}
}
```

tan_outhousing Annual N flux as TAN out of the housing indcluding N remained in biotrickling filter.

```
if ( Val(housing_type,Housing::Type) eq 'Outdoor' ) {
0;
} else {
if ( Val(air_scrubber,Housing::AirScrubber) eq 'biotrickling' ) {
# n in biotrickling vanishes
Out(tan_into_housing) - Out(nh3_nhousing) - Out(n_air_scrubber);
} else {
# n in acid scrubber adds 100% to flux into storage
Out(tan_into_housing) - Out(nh3_nhousing);
}
}
```

share_tan_out Share of TAN for interface to storage, the same share is assumed for solid and liquid parts of output. Mineralization and immobilization are considerd in the storage module.

```
if(  Out(n_outhousing) != 0 ){
Out(tan_outhousing) / Out(n_outhousing)
}else{0};
```

n_outhousing_liquid Annual N flux out of housing, slurry or liquid fraction from fattening pigs.

```
if(Val(housing_type,Housing::Type) eq 'Outdoor'){
0
}else {
Out(n_outhousing) * Val(share_liquid,Housing::Type)
};
```

tan_outhousing_liquid Annual N flux as TAN out of housing, slurry or liquid fraction from fattening pigs.

```
if(Val(housing_type,Housing::Type) eq 'Outdoor'){
0
}else {
Out(n_outhousing) * Val(share_liquid,Housing::Type) * Out(share_tan_out)
};
```

n_outhousing_solid Annual N flux out of housing, manure fraction of N flux from fattening pigs.

```
if(Val(housing_type,Housing::Type) eq 'Outdoor'){
0
}else {
Out(n_outhousing) * ( 1 - Val(share_liquid,Housing::Type) )
};
```

tan_outhousing_solid Annual N flux as TAN out of housing, manure fraction of N flux from fattening pigs.

```
if(Val(housing_type,Housing::Type) eq 'Outdoor'){
0
}else {
```

```
Out(n_outhousing) * ( 1 - Val(share_liquid,Housing::Type) ) * Out(share_tan_out)
};
```

1.57 Livestock::FatteningPigs::Housing::CFreeFactor

This process selects the correction factor for the specific housing types for fattening pigs. Among these correction factors are the emission rate for the housing type, the correction factor for the housing type area, the correction factor for the area per animal, the liquid and solid share for the housing type as well as the factor on what barn size increases the regularized minimal (limited to 0.5).

1.57.1 Inputs

free_correction_factor Factor to define free.

1.57.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a fattening pigs housing of "
. In(free_correction_factor) . "\%.\n",
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Stall der Mastschweine von "
. In(free_correction_factor) . "\% eingegeben.\n",
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions dans les "
. "stabulations pour porcs à l'engrais de " .  In(free_correction_factor) . "\%.\n"
});
return 1 - In(free_correction_factor)/100;
}
else {
return 1;
}
```

1.58 Livestock::FatteningPigs::Housing::MitigationOptions

This submodul calculates the annual NH3 reduction due to an air exhaust scrubber in fattening pig housing systems according to the UNECE guideline 2007.

1.58.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.58.2 Inputs

mitigation_housing_floor UNECE housing tasks for partly slatted floor: with scraper (concrete_-slats), with flush channels(no areation), with flush channels(areation), with flush gutters/tubes (no areation), with flush gutters/tubes (areation), with channels slanted walls(concrete slats), with channel slanted walls (metal slats), with scraper (metal slats).

mitigation_housing_floor_LU Housing tasks for partly slatted floor: with scraper (concrete_-slats), with flush channels(no areation), with flush channels(areation), with flush gutters/tubes (no areation), with flush gutters/tubes (areation), with channels slanted walls(concrete slats), with channel slanted walls (metal slats), with scraper (metal slats).

mitigation_housing_air Mitigation option air supply for pigs

1.58.3 Outputs

red_housing_floor_SHL Reduction factor for the emission due to UNECE housing systems tasks for fully and partly slatted floors.

```
given ( In(mitigation_housing_floor) ) {
when "with_scraper_concrete_slats" {
Tech(red_PSF_with_scraper_concrete_slats);
}
when "with_flush_channels_no_areation" {
Tech(red_PSF_with_flush_channels_no_areation);
}
when "with_flush_channels_areation" {
Tech(red_PSF_with_flush_channels_areation);
}
when "with_flush_gutters_tubes_no_areation" {
Tech(red_PSF_with_flush_gutters_tubes_no_areation);
}
when "with_flush_gutters_tubes_areation" {
Tech(red_PSF_with_flush_gutters_tubes_areation);
}
when "with_channels_slanted_walls_concrete_slats" {
Tech(red_PSF_with_channels_slanted_walls_concrete_slats);
}
when "with_channel_slanted_walls_metal_slats" {
Tech(red_PSF_with_channel_slanted_walls_metal_slats);
}
when "with_scraper_metal_slats" {
Tech(red_PSF_with_scraper_metal_slats);
}
when "none" {
0;
}
default {
# Warning falls not defined mitigation_housing_floor_LU
0;
}
}
```

red_housing_floor_LU Reduction factor for the emission due to housing systems tasks.

```
return 0 unless not defined In(mitigation_housing_floor);
given ( In(mitigation_housing_floor_LU) ) {
when "with_scraper_concrete_slats" {
Tech(red_PSF_with_scraper_concrete_slats);
}
when "with_flush_channels_no_areation" {
Tech(red_PSF_with_flush_channels_no_areation);
}
when "with_flush_gutters_tubes_no_areation" {
Tech(red_PSF_with_flush_gutters_tubes_no_areation);
}
when "with_scraper_metal_slats" {
Tech(red_PSF_with_scraper_metal_slats);
}
when "with_channels_slanted_walls_concrete_slats" {
Tech(red_PSF_with_channels_slanted_walls_concrete_slats);
}
```

```
when "with_channel_slanted_walls_metal_slats" {
Tech(red_PSF_with_channel_slanted_walls_metal_slats);
}
when "none" {
0;
}
default {
# Warning
0;
}
}
```

red_housing_floor Capture floor mitigation for either model.

```
Out(red_housing_floor_SHL) + Out(red_housing_floor_LU);
```

red_housing_air Reduction factor for the emission due to the use of housing system adaptations.

```
if (In(mitigation_housing_air)  eq 'low_impuls_air_supply'){
return( Tech(red_low_impuls_air_supply));
}
else {
return 0;
}
```

1.59 Livestock::FatteningPigs::Housing::AirScrubber

This submodul calculates the annual NH3 reduction due to an exhaust air scrubber in fattening pig housing systems according to the UNECE guideline 2007.

1.59.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.59.2 Inputs

air_scrubber Air exhaust scrubber (none, acid, biotrickling_filter).

1.59.3 Outputs

air_scrubber air exhaust scrubber in housing systems.

```
In(air_scrubber);
```

red_air_scrubber Reduction factor for the emission due to the use of an air exhaust scrubber in housing systems.

```
given ( In(air_scrubber) ) {
when 'acid' {
return Tech(red_acid_air_scrubber);
}
when 'biotrickling' {
return Tech(red_biotrickling_filter_air_scrubber);
}
when 'none' {
return 0;
}
```

```
default {
writeLog({
en => "Invalid 'exhaust air scrubber' entered!",
de => "Ungültige 'Abluftreinigung' eingegeben!",
fr => "Entrée non valable pour 'Traitement des effluents gazeux'!"
});
return 0;
}
}
```

1.60 Livestock::FatteningPigs::Housing::KArea

It is assumed that additional areas in stable, as example if the stabel is built for more animal, are fully used. So the emissions increase 5
the double of the regularized minimal area.

TODO (All): Provide Articles which confirm/justify the assumption. As decided on workshop from 2.11.2007.

1.60.1 Inputs

dimensioning_barn Number of available animal places.

1.60.2 Outputs

dimensioning_barn Number of anymal places (needed in other modules).

```
In(dimensioning_barn);
```

area_increase Factor on what barn size does increase the regularized minimal, limited to 0.5

```
if ( (Val(animals, ..::Excretion) < In(dimensioning_barn) ) and (Val(animals, ..::Excretion) != 0) ) {
if ( In(dimensioning_barn) >= (Val(animals, ..::Excretion) * 1.5) ) {
0.5;
}
else {
In(dimensioning_barn) / Val(animals, ..::Excretion) - 1;
}
}
else {
0.0;
}
```

c_area Correction factor for area per animal.

```
1 + (Out(area_increase) * Tech(k_area));
```

1.61 Livestock::FatteningPigs::Grazing

This process calculates the annual NH3 emission from grazing fattening pigs based on the N excreted on the pastures. The N excreted is calculated according to the time the animals spend on the pasture. It assumes that the excretions on the pasture are proportional to the time the animals are grazed.

1.61.1 References:

Bussink DW 1992. Ammonia volatilization from grassland receiving nitrogen fertilizer and rotationally grazed by dairy cattle. Fertilizer Research 33:257-265.

Bussink DW 1994. Relationship between ammonia volatilization and nitrogen fertilizer application rate, intake and excretion of herbage nitrogen by cattle on grazed swards. Fertilizer Research 38:111-121.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

Jarvis SC, Hatch DJ, Lockyer DR 1989. Ammonia fluxes from grazed grassland: annual losses from cattle production systems and their relation to nitrogen inputs. Journal of Agricultural Science 113:99-108.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Ross CA, Jarvis SC 2001. Measurement of emission and deposition pattern of ammonia from urine in grass swards. Atmospheric environment 35:867-875.

1.61.2 Inputs

ignore Just a work around for modules without input parameters.

1.61.3 Outputs

n_into_grazing Annual N excretion during grazing for fattening pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
return Val(n_excretion, Excretion);
}else {
return 0;
}
```

tan_into_grazing Annual soluble N (TAN) excretion during grazing for fattening pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
return Val(tan_excretion, Excretion);
}else {
return 0;
}
```

ef_nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
Tech(er_fattening_pig_grazing);
}else {
0;
}
```

nh3_ngrazing Annual NH3 emission from all fattening pigs from grazing.

```
Out(tan_into_grazing) * Out(ef_nh3_ngrazing);
```

n2_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2_fattening_pig_grazing);
```

no_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_no_fattening_pig_grazing);
```

n2o_ngrazing Annual total N2O emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2o_fattening_pig_grazing);
```

n_remain_grazing Annual N input on pastures.

```
Out(n_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

tan_remain_grazing Annual N input on pastures.

```
Out(tan_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

1.62 Livestock::FatteningPigs::Housing::Type

This process selects the correction factor for the specific housing types for fattening pigs. Among these correction factors are the emission rate for the housing type, the correction factor for the housing type area, the correction factor for the area per animal, the liquid and solid share for the housing type as well as the factor on what barn size increases the regularized minimal (limited to 0.5).

1.62.1 Inputs

housing_type Type of housing.

1.62.2 Outputs

housing_type Housing type (needed in other modules).

```
In(housing_type);
```

er_housing Emission rate for the housing type.

```
if ( Out(housing_type) eq 'Outdoor' ) {
return Val(er_housing,Type::Outdoor);
}
elsif ( Out(housing_type) eq 'Slurry_Conventional' ) {
return Val(er_housing,Type::Slurry_Conventional);
}
elsif ( Out(housing_type) eq 'Slurry_Label' ) {
return Val(er_housing,Type::Slurry_Label);
}
elsif ( Out(housing_type) eq 'Slurry_Label_Open' ) {
return Val(er_housing,Type::Slurry_Label_Open);
}
elsif ( Out(housing_type) eq 'Deep_Litter' ) {
return Val(er_housing,Type::Deep_Litter);
}
```

share_liquid Liquid share for the housing type.

```
if ( Out(housing_type) eq 'Outdoor' ) {
return Val(share_liquid,Type::Outdoor);
}
elsif ( Out(housing_type) eq 'Slurry_Conventional' ) {
return Val(share_liquid,Type::Slurry_Conventional);
```

```
}
elsif ( Out(housing_type) eq 'Slurry_Label' ) {
return Val(share_liquid,Type::Slurry_Label);
}
elsif ( Out(housing_type) eq 'Slurry_Label_Open' ) {
return Val(share_liquid,Type::Slurry_Label_Open);
}
elsif ( Out(housing_type) eq 'Deep_Litter' ) {
return Val(share_liquid,Type::Deep_Litter);
}
```

share_solid Solid share for the housing type.

```
if ( Out(housing_type) eq 'Outdoor' ) {
return Val(share_solid,Type::Outdoor);
}
elsif ( Out(housing_type) eq 'Slurry_Conventional' ) {
return Val(share_solid,Type::Slurry_Conventional);
}
elsif ( Out(housing_type) eq 'Slurry_Label' ) {
return Val(share_solid,Type::Slurry_Label);
}
elsif ( Out(housing_type) eq 'Slurry_Label_Open' ) {
return Val(share_solid,Type::Slurry_Label_Open);
}
elsif ( Out(housing_type) eq 'Deep_Litter' ) {
return Val(share_solid,Type::Deep_Litter);
}
```

share_indoor Factor for considering indoor mitigation efficiencies.

```
if ( (Out(housing_type) eq 'Slurry_Label') or (Out(housing_type) eq 'Slurry_Label_Open') ) {
return 0.5;
}
else {
return 1;
}
```

1.63 Livestock::FatteningPigs::Housing::Type::Outdoor

This process describes the correction factors for outdoor fattening pigs such as the housing specific emission rate, the liquid share and solid share. Outdoor fattening pigs do not have any housing emissions, as everything is excreted on pasture.

TODO (Note): justification

1.63.1 Inputs

ignore Just a work around for modules without input parameters.

1.63.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid For the outdoor fattening pigs 0% of the manure goes into the solid fraction for storage/application.

```
0
```

1.64 Livestock::FatteningPigs::Housing::Type::Deep_Litter

This process describes the correction factors for the label deep litter fattening pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.64.1 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.65 Livestock::FatteningPigs::Housing::Type::Slurry_Label_Open

This process describes the correction factors for the label slurry open fattening pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.65.1 Inputs

ignore Just a work around for modules without input parameters.

1.65.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.66 Livestock::FatteningPigs::Housing::Type::Slurry_Label

This process describes the correction factors for the label slurry fattening pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.66.1 Inputs

ignore Just a work around for modules without input parameters.

1.66.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.67 Livestock::FatteningPigs::Housing::Type::Slurry_Conventional

This process describes the correction factors for the conventional slurry fattening pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.67.1 Inputs

ignore Just a work around for modules without input parameters.

1.67.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.68 Livestock::FatteningPigs::Excretion

This process calculates the annual N excretion (total N and Nsol) of fattening pigs according to the crude protein and energy content of the feed ration.

TODO (Harald Menzi): Formulation of Beat Reidy maybe mistaken,

1.68.1 References:

BLW, SRVA, LBL 2003. Weisungen zur Beruecksichtigung von Ökofuttern in der Suisse-Bilanz. 2003.

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Petersen SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.68.2 Inputs

animalcategory Animal category

fattening_pigs Number of places for fattening pigs for the selected type in barn.

inp_n_excretion Annual standard N excretion for a fattening pig

feeding_phase_1_crude_protein Bei Durchmast (von 25 kg bis zur Schlachtung gleiches Futter eingesetzt): den RP Gehalt des verwendeten Durchmastfutters eingeben und unten bei Mastphase 2 und 3 den gleichen Wert wie für Mastphase 1 eingeben.

Bei Phasenfütterung: RP-Gehalt des auf dem Betrieb verwendeten Futters von Mastphase 1 eingeben.

<p>Standardwerte zur Eingabe, falls die Gehalte des auf dem Betrieb verwendeten Futters nicht bekannt sind:</p>

 Bei Durchmast: 170 g RP /kg

Mastphase 1 bei 2-Phasenfütterung und bei 3-Phasenfütterung: 175 g RP /kg

<p>Bei Verwendung von NPr Futter:</p> Bei Durchmast: 155 g RP /kg

Mastphase 1 bei 2-Phasenfütterung und bei 3-Phasenfütterung: 160 g RP /kg

feeding_phase_2_crude_protein Bei Durchmast: den gleichen Wert wie für Mastphase 1 eingeben

Bei 2-Phasenfütterung: den RP Gehalt des auf dem Betrieb verwendeten Futters von Mastphase 2 eingeben und unten bei Mastphase 3 den gleichen Wert wie für Mastphase 2 eingeben

Bei 3-Phasenfütterung: den RP Gehalt des auf dem Betrieb verwendeten Futters von Mastphase 2 eingeben

<p>Vorschlag für Standardwerte:</p>

 Mastphase 2 bei 2-Phasenfütterung und bei 3-Phasenfütterung: 160 g RP /kg

<p>Bei Verwendung von NPr Futter:</p>

 Mastphase 2 bei 2-Phasenfütterung und bei 3-Phasenfütterung: 150 g RP /kg

feeding_phase_3_crude_protein Bei Durchmast und 2-Phasenfütterung: den gleichen Wert wie für Mastphase 1 eingeben

Bei 3-Phasenfütterung: den RP Gehalt des auf dem Betrieb verwendeten Futters von Mastphase 3 eingeben

<p>Standardwerte zur Eingabe, falls die Gehalte des auf dem Betrieb verwendeten Futters nicht bekannt sind:</p>

 Mastphase 3 bei 3-Phasenfütterung: 150 g RP /kg

<p>Bei Verwendung von NPr Futter:</p>

 Mastphase 3 bei 3-Phasenfütterung: 140 g RP /kg

energy_content Energy content of feed ration.

1.68.3 Outputs

animals Number of fattening pigs of a specific category.

```
In(fattening_pigs);
```

animalcategory Animal category

```
In(animalcategory);
```

crude_protein Crude protein content of feed ration - for 1-, 2- or 3-phase-feeding.

```
if ( In(feeding_phase_3_crude_protein) == 0 or
In(feeding_phase_2_crude_protein) == In(feeding_phase_3_crude_protein) ) {
if ( In(feeding_phase_2_crude_protein) == 0 ) {
In(feeding_phase_1_crude_protein);
}
else {
In(feeding_phase_1_crude_protein) * Tech(phase_1_2_duration) +
In(feeding_phase_2_crude_protein) * Tech(phase_2_2_duration);
}
}
else {
In(feeding_phase_1_crude_protein) * Tech(phase_1_3_duration) +
In(feeding_phase_2_crude_protein) * Tech(phase_2_3_duration) +
In(feeding_phase_3_crude_protein) * Tech(phase_3_3_duration)
}
```

n_excretion_animal Annual standard N excretion for fattening pigs according to Walther et al. (2001).

```
if ( lc In(inp_n_excretion) eq 'standard' ) {
my $exc = Tech(standard_N_excretion_fattening_pigs) *
(1 - (
Tech(standard_crude_protein_fattening_pigs) -
(Out(crude_protein) * Tech(standard_energy_content_fattening_pigs) / In(energy_content))
) *
Tech(cfeed_fattening_pigs));
if ( $exc < Tech(minimal_N_excretion_fattening_pigs) ) {
writeLog({
en => "The entry for the N excretion is below the minimum for fattening pigs. \n"
. "The excretion was set to " . Tech(minimal_N_excretion_fattening_pigs) . " kg N per animal!",
de => "Die gewählte N Ausscheidung liegt unterhalb des Minimums für Mastschweine! \n"
. "Die Ausscheidung wurde auf ". Tech(minimal_N_excretion_fattening_pigs) ." kg N pro Tier gesetzt!",
fr => "FIX: Les teneurs choisies en énergie et en matière azotée de la ration provoquent une "
."excrétion azotée inférieure au minimum prévu pour les porcs à l'engrais. "
."L'excrétion a été fixée à ". Tech(minimal_N_excretion_fattening_pigs) ."kg N par animal. \n"
});
return Tech(minimal_N_excretion_fattening_pigs);
} else {
return $exc;
}
} else {
if ( In(inp_n_excretion) < Tech(minimal_N_excretion_fattening_pigs) ) {
writeLog({
en => "The entry for the N excretion is below the minimum for fattening pigs. \n"
. "The excretion was set to " . Tech(minimal_N_excretion_fattening_pigs) . " kg N per animal!",
de => "Die gewählte N Ausscheidung liegt unterhalb des Minimums für Mastschweine! \n"
. "Die Ausscheidung wurde auf ". Tech(minimal_N_excretion_fattening_pigs) ." kg N pro Tier gesetzt!",
fr => "FIX: Les teneurs choisies en énergie et en matière azotée de la ration provoquent une "
."excrétion azotée inférieure au minimum prévu pour les porcs à l'engrais. "
."L'excrétion a été fixée à ". Tech(minimal_N_excretion_fattening_pigs) ."kg N par animal. \n"
});
return Tech(minimal_N_excretion_fattening_pigs);
} else {
if ( (In(inp_n_excretion) < 0.25 * Tech(standard_N_excretion_fattening_pigs)) or (In(inp_n_excretion) > 1.75
writeLog({
en => "FIX: Die eingegebene N-Ausscheidung für Mastschweine weicht mehr als 75% vom Standard ab.",
```

```
de => "Die eingegebene N-Ausscheidung für Mastschweine weicht mehr als 75% vom Standard ab.",
fr => "FIX: Die eingegebene N-Ausscheidung für Mastschweine weicht mehr als 75% vom Standard ab.",
});
}
return In(inp_n_excretion);
}

}
```

n_excretion Annual total N excreted by a specified number of fattening pigs.

```
Out(n_excretion_animal) * Out(animals);
```

tan_excretion Annual soluble N excreted by a specified number of fattening pigs.

```
Tech(share_Nsol) * Out(n_excretion);
```

1.69 Livestock::Pig

This process summarizes the annual emission of NH3 for pigs from housing and grazing. And it summarizes the annual N flux from housing to the storage for pigs.

1.69.1 Outputs

n_excretion Annual N excreted by pigs.

```
Val(n_excretion, Pig::Excretion);
```

n_from_pig Annual N flux from pigs housing.

```
Val(n_outhousing, Pig::Housing);
```

tan Annual N flux as TAN from pigs housing.

```
Val(tan_outhousing, Pig::Housing);
```

n_liquid_from_pig Annual N flux from pigs housing, liquid fraction.

```
Val(n_outhousing_liquid, Pig::Housing);
```

tan_liquid Annual N flux as TAN from pigs housing, liquid fraction.

```
Val(tan_outhousing_liquid, Pig::Housing);
```

n_solid_from_pig Annual N flux from pigs housing, solid fraction.

```
Val(n_outhousing_solid, Pig::Housing);
```

tan_solid Annual N flux as TAN from pigs housing, liquid fraction.

```
Val(tan_outhousing_solid, Pig::Housing);
```

nh3_npig Annual NH3 emission from pigs housing and grazing (production).

```
Val(nh3_nhousing, Pig::Housing) +
Val(nh3_ngrazing, Pig::Grazing);

########### Other Voltaile N Flux
```

er_n2_nsolid Annual N2 emissions from pigs housing, yard and grazing (production).

```
given ( Val(housing_type, Pig::Housing::Type) ) {
return Tech(er_n2_solid_Slurry)        when 'Slurry_Conventional';
return Tech(er_n2_solid_Slurry)        when 'Slurry_Label';
return Tech(er_n2_solid_Solid)         when 'Deep_Litter';
return Tech(er_n2_solid_Solid)         when 'Outdoor';
default { return 0; }
};
```

er_no_nsolid Annual N2 emissions from pigs housing, yard and grazing (production).

```
given ( Val(housing_type, Pig::Housing::Type) ) {
return Tech(er_no_solid_Slurry)         when 'Slurry_Conventional';
return Tech(er_no_solid_Slurry)         when 'Slurry_Label';
return Tech(er_no_solid_Solid)          when 'Deep_Litter';
return Tech(er_no_solid_Solid)          when 'Outdoor';
default { return 0; }
};
```

er_n2o_nsolid Annual N2 emissions from pigs housing, yard and grazing (production).

```
given ( Val(housing_type, Pig::Housing::Type) ) {
return Tech(er_n2o_solid_Slurry)         when 'Slurry_Conventional';
return Tech(er_n2o_solid_Slurry)         when 'Slurry_Label';
return Tech(er_n2o_solid_Solid)          when 'Deep_Litter';
return Tech(er_n2o_solid_Solid)          when 'Outdoor';
default { return 0; }
};
```

er_n2_nliquid Annual N2 emissions from pigs housing, yard and grazing (production).

```
given ( Val(housing_type, Pig::Housing::Type) ) {
return Tech(er_n2_liquid_Slurry)         when 'Slurry_Conventional';
return Tech(er_n2_liquid_Slurry)         when 'Slurry_Label';
return Tech(er_n2_liquid_Solid)          when 'Deep_Litter';
return Tech(er_n2_liquid_Solid)          when 'Outdoor';
default { return 0; }
};
```

er_no_nliquid Annual N2 emissions from pigs housing, yard and grazing (production).

```
given ( Val(housing_type, Pig::Housing::Type) ) {
return Tech(er_no_liquid_Slurry)         when 'Slurry_Conventional';
return Tech(er_no_liquid_Slurry)         when 'Slurry_Label';
return Tech(er_no_liquid_Solid)          when 'Deep_Litter';
return Tech(er_no_liquid_Solid)          when 'Outdoor';
default { return 0; }
};
```

er_n2o_nliquid Annual N2 emissions from pigs housing, yard and grazing (production).

```
given ( Val(housing_type, Pig::Housing::Type) ) {
return Tech(er_n2o_liquid_Slurry)         when 'Slurry_Conventional';
return Tech(er_n2o_liquid_Slurry)         when 'Slurry_Label';
return Tech(er_n2o_liquid_Solid)          when 'Deep_Litter';
return Tech(er_n2o_liquid_Solid)          when 'Outdoor';
default { return 0; }
};
```

n2_nsolid Annual N2 emissions from pigs housing, yard and grazing (production).

```
( Val(n_into_housing, Pig::Housing) * (1-Val(share_liquid, Pig::Housing::Type) )
) * Out(er_n2_nsolid);
```

no_nsolid_pig Annual NO emissions from pigs housing, yard and grazing (production).

```
( Val(n_into_housing, Pig::Housing) * (1-Val(share_liquid, Pig::Housing::Type) )
) * Out(er_no_nsolid);
```

n2o_nsolid Annual N2O emissions from pigs housing, yard and grazing (production).

```
( Val(n_into_housing, Pig::Housing) * (1-Val(share_liquid, Pig::Housing::Type) )
) * Out(er_n2o_nsolid);
```

n2_nliquid Annual N2 emissions from pigs housing, yard and grazing (production).

```
( Val(n_into_housing, Pig::Housing) * Val(share_liquid, Pig::Housing::Type)
) * Out(er_n2_nliquid);
```

no_nliquid_pig Annual NO emissions from pigs housing, yard and grazing (production).

```
( Val(n_into_housing, Pig::Housing) * Val(share_liquid, Pig::Housing::Type)
) * Out(er_no_nliquid);
```

n2o_nliquid Annual N2O emissions from pigs housing, yard and grazing (production).

```
( Val(n_into_housing, Pig::Housing) * Val(share_liquid, Pig::Housing::Type)
) *  Out(er_n2o_nliquid);
```

1.70 Livestock::Pig::Housing

This process calculates the NH3 emission in pig housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emissions, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.70.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.70.2 Outputs

n_into_housing Annual N flux into the house.

```
Val(n_excretion, Excretion) - Val(n_into_grazing, Grazing);
```

tan_into_housing Annual N flux as TAN into the house.

```
Val(tan_excretion, Excretion) - Val(tan_into_grazing, Grazing);
```

nh3_nhousing Annual NH3 emission from pig housing systems.

```
Out(tan_into_housing) *
Val(er_housing, Housing::Type) *
Val(c_free_factor_housing, Housing::CFreeFactor) *
Val(c_area, Housing::KArea) *
(
# indoor part + mitigation
Val(share_indoor, Housing::Type) *
(1 - Val(red_air_scrubber, Housing::AirScrubber)) *
(1 - Val(red_housing_floor, Housing::MitigationOptions)) *
(1 - Val(red_housing_air, Housing::MitigationOptions)) +
# outdoor part + mitigation
(1 - Val(share_indoor, Housing::Type))
);
```

n_air_scrubber Annual N of NH3 emission remaining in air scrubber from pig housing systems.

```
Val(red_air_scrubber, Housing::AirScrubber) *
# multiplied with indoor loss without air scrubber mitigation
Val(tan_excretion,Excretion) *
Val(er_housing, Housing::Type) *
Val(c_area, Housing::KArea) *
Val(c_free_factor_housing, Housing::CFreeFactor) *
Val(share_indoor, Housing::Type) *
(1 - Val(red_housing_floor, Housing::MitigationOptions)) *
(1 - Val(red_housing_air, Housing::MitigationOptions));
```

n_outhousing Annual N flux out of the housing including N remained in biotrickling filter.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
0
}else {
if(Val(air_scrubber, Housing::AirScrubber) eq 'biotrickling'){
Out(n_into_housing) - Out(nh3_nhousing) - Out(n_air_scrubber)
}else{
Out(n_into_housing) - Out(nh3_nhousing)
};
};
```

tan_outhousing Annual N flux as TAN out of the housing indcluding N remained in biotrickling filter. (without remains from acid filter)

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
0
}else {
if(Val(air_scrubber, Housing::AirScrubber) eq 'biotrickling'){
Out(tan_into_housing) - Out(nh3_nhousing) - Out(n_air_scrubber)
}else{
Out(tan_into_housing) - Out(nh3_nhousing)
};
};
```

share_tan_out Share of TAN for interface to storage, the same share is assumed for solid and liquid parts of output. Mineralization and immobilization are considerd in the storage module.

```
if(  Out(n_outhousing) != 0 ){
Out(tan_outhousing) / Out(n_outhousing)
}else{0};
```

n_outhousing_liquid Annual N flux out of housing, slurry or liquid fraction from pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
0
}else {
Out(n_outhousing) * Val(share_liquid, Housing::Type)
};
```

tan_outhousing_liquid Annual N flux as TAN out of housing, slurry or liquid fraction from pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
0
}else {
Out(n_outhousing) * Val(share_liquid, Housing::Type) * Out(share_tan_out)
};
```

n_outhousing_solid Annual N flux out of housing, manure fraction of N flux from pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
0
}else {
Out(n_outhousing) * ( 1 - Val(share_liquid, Housing::Type) )
};
```

tan_outhousing_solid Annual N flux as TAN out of housing, manure fraction of N flux from pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
0
}else {
Out(n_outhousing) * ( 1 - Val(share_liquid, Housing::Type) ) * Out(share_tan_out)
};
```

1.71 Livestock::Pig::Housing::CFreeFactor

This process selects the correction factor for the specific housing types for pigs. Among these correction factors are the emission rate for the housing type, the correction factor for the housing type area, the correction factor for the area per animal, the liquid and solid share for the housing type as well as the factor on what barn size increases the regularized minimal (limited to 0.5).

1.71.1 Inputs

free_correction_factor Factor to define free.

1.71.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a housing of the category pigs of "
. In(free_correction_factor) . "\%.\n",
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen  Stall der Kategorie Schweine von "
. In(free_correction_factor) . "\% eingegeben.\n",
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions dans les "
. "stabulations pour porcs de " . In(free_correction_factor) . "\%.\n"
});
return 1 - In(free_correction_factor) / 100;
}
else {
return 1;
}
```

1.72 Livestock::Pig::Housing::MitigationOptions

This submodul calculates the annual NH3 reduction due to an exhaust air scrubber in housing systems according to the UNECE guideline 2007.

1.72.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.72.2 Inputs

mitigation_housing_floor UNECE housing tasks for partly slatted floor: with scraper (concrete_-slats), with flush channels(no areation), with flush channels(areation), with flush gutters/tubes (no areation), with flush gutters/tubes (areation), with channels slanted walls(concrete slats), with channel slanted walls (metal slats), with scraper (metal slats).

mitigation_housing_floor_LU Housing tasks for partly slatted floor: with scraper (concrete_-slats), with flush channels(no areation), with flush channels(areation), with flush gutters/tubes (no areation), with flush gutters/tubes (areation), with channels slanted walls(concrete slats), with channel slanted walls (metal slats), with scraper (metal slats).

mitigation_housing_air Mitigation option air supply for pigs

1.72.3 Outputs

red_housing_floor_SHL Reduction factor for the emission due to UNECE housing systems tasks for fully and partly slatted floors.

```
given ( In(mitigation_housing_floor) ) {
when "with_scraper_concrete_slats" {
Tech(red_PSF_with_scraper_concrete_slats);
}
when "with_flush_channels_no_areation" {
Tech(red_PSF_with_flush_channels_no_areation);
}
when "with_flush_channels_areation" {
Tech(red_PSF_with_flush_channels_areation);
}
when "with_flush_gutters_tubes_no_areation" {
Tech(red_PSF_with_flush_gutters_tubes_no_areation);
}
when "with_flush_gutters_tubes_areation" {
Tech(red_PSF_with_flush_gutters_tubes_areation);
}
when "with_channels_slanted_walls_concrete_slats" {
Tech(red_PSF_with_channels_slanted_walls_concrete_slats);
}
when "with_channel_slanted_walls_metal_slats" {
Tech(red_PSF_with_channel_slanted_walls_metal_slats);
}
when "with_scraper_metal_slats" {
Tech(red_PSF_with_scraper_metal_slats);
}
when "none" {
0;
}
default {
# Warning falls not defined mitigation_housing_floor_LU
0;
}
}
```

red_housing_floor_LU Reduction factor for the emission due to housing systems tasks.

```
return 0 unless not defined In(mitigation_housing_floor);
given ( In(mitigation_housing_floor_LU) ) {
when "with_scraper_concrete_slats" {
Tech(red_PSF_with_scraper_concrete_slats);
}
when "with_flush_channels_no_areation" {
Tech(red_PSF_with_flush_channels_no_areation);
}
when "with_flush_gutters_tubes_no_areation" {
Tech(red_PSF_with_flush_gutters_tubes_no_areation);
}
when "with_scraper_metal_slats" {
Tech(red_PSF_with_scraper_metal_slats);
}
when "with_channels_slanted_walls_concrete_slats" {
Tech(red_PSF_with_channels_slanted_walls_concrete_slats);
}
when "with_channel_slanted_walls_metal_slats" {
Tech(red_PSF_with_channel_slanted_walls_metal_slats);
}
when "none" {
0;
}
default {
# Warning
0;
```

```
}
}
```

red_housing_floor Capture floor mitigation for either model.

```
Out(red_housing_floor_SHL) + Out(red_housing_floor_LU);
```

red_housing_air Reduction factor for the emission due to the use of housing system adaptations.

```
if (In(mitigation_housing_air)  eq 'low_impuls_air_supply'){
return( Tech(red_low_impuls_air_supply));
}
else {
return 0;
}
```

1.73 Livestock::Pig::Housing::AirScrubber

This submodul calculates the annual NH3 reduction due to an exhaust air scrubber in housing systems according to the UNECE guideline 2007.

1.73.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.73.2 Inputs

air_scrubber Exhaust air scrubber: none, acid, biotrickling_filter.

1.73.3 Outputs

air_scrubber air exhaust scrubber in housing systems.

```
In(air_scrubber);
```

red_air_scrubber Reduction factor for the emission due to the use of an air exhaust scrubber in housing systems.

```
given ( In(air_scrubber) ) {
when 'acid' {
return Tech(red_acid_air_scrubber);
}
when 'biotrickling' {
return Tech(red_biotrickling_filter_air_scrubber);
}
when 'none' {
return 0;
}
default {
writeLog({
en => "Invalid 'exhaust air scrubber' entered!",
de => "Ungültige 'Abluftreinigung' eingegeben!",
fr => "Entrée non valable pour 'Traitement des effluents gazeux'!"
});
return 0;
}
}
```

1.74 Livestock::Pig::Housing::KArea

It is assumed that additional areas in stable, as example if the stabel is built for more animal, are fully used. So the emissions increase 5
the double of the regularized minimal area.

TODO (All): Provide Articles which confirm/justify the assumption. As decided on workshop from 2.11.2007.

1.74.1 Inputs

dimensioning_barn Number of available animal places.

1.74.2 Outputs

dimensioning_barn Number of anymal places (needed in other modules).

```
In(dimensioning_barn);
```

area_increase Factor on what barn size does increase the regularized minimal, limited to 0.5

```
if ( (Val(animals, ..::Excretion) < In(dimensioning_barn) ) and (Val(animals, ..::Excretion) != 0) ) {
if ( In(dimensioning_barn) >= (Val(animals, ..::Excretion) * 1.5) ) {
0.5;
}
else {
In(dimensioning_barn) / Val(animals, ..::Excretion) - 1;
}
}
else {
0.0;
}
```

c_area Correction factor for area per animal.

```
1 + (Out(area_increase) * Tech(k_area));
```

1.75 Livestock::Pig::Grazing

This process calculates the annual NH3 emission from grazing pigs based on the N excreted on the pastures. The N excreted is calculated according to the time the animals spend on the pasture. It assumes that the excretions on the pasture are proportional to the time the animals are grazed.

1.75.1 References:

Bussink DW 1992. Ammonia volatilization from grassland receiving nitrogen fertilizer and rotationally grazed by dairy cattle. Fertilizer Research 33:257-265.

Bussink DW 1994. Relationship between ammonia volatilization and nitrogen fertilizer application rate, intake and excretion of herbage nitrogen by cattle on grazed swards. Fertilizer Research 38:111-121.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

Jarvis SC, Hatch DJ, Lockyer DR 1989. Ammonia fluxes from grazed grassland: annual losses from cattle production systems and their relation to nitrogen inputs. Journal of Agricultural Science 113:99-108.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Ross CA, Jarvis SC 2001. Measurement of emission and deposition pattern of ammonia from urine in grass swards. Atmospheric environment 35:867-875.

Sommer SG, Sogaard HT, Moller HB, Morsing S 2001. Ammonia volatilization from sows on grassland. Atmospheric environment 35:2023-2032.

1.75.2 Outputs

n_into_grazing Annual N excretion during grazing for pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
Val(n_excretion, Excretion);
}else {
0;
}
```

tan_into_grazing Annual soluble N (TAN) excretion during grazing for pigs.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
Val(tan_excretion, Excretion);
}else {
0;
}
```

ef_nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
if(Val(housing_type, Housing::Type) eq 'Outdoor'){
Tech(er_pig_grazing);
}else {
0;
}
```

nh3_ngrazing Annual NH3 emission from all pigs from grazing.

```
Out(tan_into_grazing) * Out(ef_nh3_ngrazing);
```

n2_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2_pig_grazing);
```

no_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_no_pig_grazing);
```

n2o_ngrazing Annual total N2O emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2o_pig_grazing);
```

n_remain_grazing Annual N input on pastures.

```
Out(n_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

tan_remain_grazing Annual N input on pastures.

```
Out(tan_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
```

```
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

1.76 Livestock::Pig::Housing::Type

This process selects the correction factor for the specific housing types for pigs. Among these correction factors are the emission rate for the housing type, the correction factor for the housing type area, the correction factor for the area per animal, the liquid and solid share for the housing type as well as the factor on what barn size increases the regularized minimal (limited to 0.5).

1.76.1 Inputs

housing_type Type of housing.

1.76.2 Outputs

housing_type Housing type (needed in other modules).

```
In(housing_type);
```

er_housing Emission rate for the housing type.

```
if ( Out(housing_type) eq 'Outdoor' ) {
return Val(er_housing,Type::Outdoor);
}
elsif ( Out(housing_type) eq 'Slurry_Conventional' ) {
return Val(er_housing,Type::Slurry_Conventional);
}
elsif ( Out(housing_type) eq 'Slurry_Label' ) {
return Val(er_housing,Type::Slurry_Label);
}
elsif ( Out(housing_type) eq 'Slurry_Label_Open' ) {
return Val(er_housing,Type::Slurry_Label_Open);
}
elsif ( Out(housing_type) eq 'Deep_Litter' ) {
return Val(er_housing,Type::Deep_Litter);
}
```

share_liquid Liquid share for the housing type.

```
if ( Out(housing_type) eq 'Outdoor' ) {
return Val(share_liquid,Type::Outdoor);
}
elsif ( Out(housing_type) eq 'Slurry_Conventional' ) {
return Val(share_liquid,Type::Slurry_Conventional);
}
elsif ( Out(housing_type) eq 'Slurry_Label' ) {
return Val(share_liquid,Type::Slurry_Label);
}
elsif ( Out(housing_type) eq 'Slurry_Label_Open' ) {
return Val(share_liquid,Type::Slurry_Label_Open);
}
elsif ( Out(housing_type) eq 'Deep_Litter' ) {
return Val(share_liquid,Type::Deep_Litter);
}
```

share_solid Solid share for the housing type.

```
if ( Out(housing_type) eq 'Outdoor' ) {
return Val(share_solid,Type::Outdoor);
}
```

```
elsif ( Out(housing_type) eq 'Slurry_Conventional' ) {
return Val(share_solid,Type::Slurry_Conventional);
}
elsif ( Out(housing_type) eq 'Slurry_Label' ) {
return Val(share_solid,Type::Slurry_Label);
}
elsif ( Out(housing_type) eq 'Slurry_Label_Open' ) {
return Val(share_solid,Type::Slurry_Label_Open);
}
elsif ( Out(housing_type) eq 'Deep_Litter' ) {
return Val(share_solid,Type::Deep_Litter);
}
```

share_indoor Factor for considering outdoor part.

```
if ( (Out(housing_type) eq 'Slurry_Label') or (Out(housing_type) eq 'Slurry_Label_Open') ){
return 0.5;
}
else {
return( 1 );
}
```

1.77 Livestock::Pig::Housing::Type::Outdoor

This process describes the correction factors for outdoor pigs such as the housing specific emission rate, the liquid share and solid share. Outdoor pigs do not have any housing emissions, as everything is excreted on pasture.

TODO (Note): justification

1.77.1 Inputs

ignore Just a work around for modules without input parameters.

1.77.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid For the outdoor pigs 0% of the manure goes into the solid fraction for storage/application.

```
0
```

1.78 Livestock::Pig::Housing::Type::Deep_Litter

This process describes the correction factors for the label deep litter pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.78.1 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.79 Livestock::Pig::Housing::Type::Slurry_Label_Open

This process describes the correction factors for the label slurry open pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.79.1 Inputs

ignore Just a work around for modules without input parameters.

1.79.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.80 Livestock::Pig::Housing::Type::Slurry_Label

This process describes the correction factors for the label slurry pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.80.1 Inputs

ignore Just a work around for modules without input parameters.

1.80.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.81 Livestock::Pig::Housing::Type::Slurry_Conventional

This process describes the correction factors for the conventional slurry pig housing system such as the housing specific emission rate, the liquid share and solid share.

1.81.1 Inputs

ignore Just a work around for modules without input parameters.

1.81.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Liquid part of Ntot for selected housing type.

```
Tech(share_liquid);
```

share_solid Solid part of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

1.82 Livestock::Pig::Excretion

This process calculates the annual N excretion (total N and Nsol) of different pig categories according to the crude protein and energy content of the feed ration.

TODO (Harald Menzi): Formulation of Beat Reidy maybe mistaken,

1.82.1 References:

BLW, SRVA, LBL 2003. Weisungen zur Beruecksichtigung von Ökofuttern in der Suisse-Bilanz. 2003.

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.82.2 Inputs

animalcategory Pig category (nursing sows, dry sows, gilts, weaned piglets (up to 25_kg), and boars).

pigs Number of places for pigs for the selected type in barn.

inp_n_excretion Annual standard N excretion for a pig

crude_protein <p>Es sind die Daten des auf dem Betrieb verwendeten Futters einzugeben.</p>

<p>Bei Verwendung von zwei verschiedenen Futtermitteln bei abgesetzten Ferkeln: nach Verzehr gemittelte Gehalt einsetzen.</p>

<p>Standardwerte zur Eingabe, falls die Gehalte des auf dem Betrieb verwendeten Futters nicht bekannt sind:</p>

 Säugende Sauen 180 g RP /kg;

Galtsauen 145 g RP /kg;

Remonten 170 g RP /kg ;

Eber 171 g RP /kg ;

abgesetzte Ferkel 177 g RP /kg

<p>RP-Gehalte von nährstoffreduziertem Futter: </p> Säugende Sauen 155 g RP /kg;

Galtsauen, Remonten und Eber 135 g RP /kg;

abgesetzte Ferkel 165 g RP /kg

energy_content <p>Es sind die Daten des auf dem Betrieb verwendeten Futters einzugeben.</p>

<p>Bei Verwendung von zwei verschiedenen Futtermitteln bei abgesetzten Ferkeln: nach Verzehr gemittelten Gehalt einsetzen.</p>

Standardwerte zur Eingabe, falls die Gehalte des auf dem Betrieb verwendeten Futters nicht bekannt sind:

 Säugende Sauen: 13.7 MJ VES /kg Galtsauen: 12.1 MJ VES /kg Remonten: 14.0 MJ VES /kg Eber: 12.9 MJ VES /kg; abgesetzte Ferkel: 13.7 MJ VES /kg

1.82.3 Outputs

animals Number of pigs of a specific category.

```
In(pigs);
```

animalcategory Animal category

```
In(animalcategory);
```

n_excretion_animal Annual total N excreted per animal.

```
my $minimal = $TE->{'minimal_N_excretion_' . Out(animalcategory)};
my $stdN = $TE->{'standard_N_excretion_' . Out(animalcategory)};
if ( lc In(inp_n_excretion) eq 'standard' ) {
my $Nexcr =  $stdN*
(1 -
($TE->{'standard_crude_protein_' . Out(animalcategory)} -
In(crude_protein) * $TE->{'standard_energy_content_' . Out(animalcategory)} / In(energy_content)
) *
$TE->{'cfeed_' . Out(animalcategory)});
if( $Nexcr < $minimal ) {
writeLog({
en => "The entries for the contents of energy and crude protein of the ration result in a N excretion below t
. "The excretion was set to " . $minimal ." kg N per animal!",
de => "Der gewählte Energie- und Rohproteingehalt der Ration bewirken eine N Ausscheidung unterhalb des Minim
. "Die Ausscheidung wurde auf ". $minimal ."  kg N pro Tier gesetzt!",
fr => "Franz.: Der gewählte Energie- und Rohproteingehalt der Ration bewirken eine N Ausscheidung unterhalb d
```

```
. "Die Ausscheidung wurde auf ". $minimal ."  kg N pro Tier gesetzt!"
});
return $minimal;
} else {
return $Nexcr;
}
} else {
if( In(inp_n_excretion) < $minimal ) {
writeLog({
en => "The entry for N excretion is below the minimum for pigs.\n"
. "The excretion was set to " . $minimal ." kg N per animal!",
de => "Der gewählte N Ausscheidung liegt unterhalb des Minimums für Schweine! \n"
. "Die Ausscheidung wurde auf ". $minimal ."  kg N pro Tier gesetzt!",
fr => "Franz.: Der gewählte N Ausscheidung liegt unterhalb des Minimums für Schweine! \n"
. "Die Ausscheidung wurde auf ". $minimal ."  kg N pro Tier gesetzt!"
});
return $minimal;
} else {
if ( In(inp_n_excretion) < 0.25 * $stdN or
In(inp_n_excretion) > 1.75 * $stdN ) {
writeLog({
en => "FIX: Die eingegebene N-Ausscheidung für Mastschweine weicht mehr als 75% vom Standard ab.",
de => "Die eingegebene N-Ausscheidung für Mastschweine weicht mehr als 75% vom Standard ab.",
fr => "FIX: Die eingegebene N-Ausscheidung für Mastschweine weicht mehr als 75% vom Standard ab.",
});
}
return In(inp_n_excretion);
}
}
```

n_excretion Annual total N excreted by a specified number of animals.

```
Out(n_excretion_animal) * Out(animals);
```

tan_excretion Annual soluble N excreted by a specified number fo pigs.

```
Tech(share_Nsol) * Out(n_excretion);
```

1.83 Livestock::OtherCattle

This process summarizes the annual emission of NH3 for cattle (apart from dairy cows) from housing, yard and grazing. Further it summarizes the annual N flux from housing and yard to the storage for cattle.

1.83.1 Inputs

ignore Just a work around for modules without input parameters.

1.83.2 Outputs

dimensioning_check Check if number of animals <= number of animal places.

```
if (  Val(dimensioning_barn, OtherCattle::Housing::KArea) < Val(animals, OtherCattle::Excretion) ) {
writeLog({en=>"The number of available animal places must be equal or higher than the number of animals.",
de=>"Anzahl Tierplätze muss grösser oder gleich der Anzahl der Tiere sein.",
fr=>"Le nombre de places dans l'étable doit être au moins égal au nombre d'animaux."
});
}
return;

########### Other Voltaile N Flux
```

er_n2_nsolid Annual N2 emissions from cattles housing, yard and grazing (production).

```
given ( Val(housing_type, OtherCattle::Housing::Type) ) {
return Tech(er_n2_solid_Slurry)                       when 'Tied_Housing_Slurry';
return Tech(er_n2_solid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_solid_Slurry)                       when 'Loose_Housing_Slurry';
return Tech(er_n2_solid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_solid_Solid)                        when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_no_nsolid Annual N2 emissions from cattles housing, yard and grazing (production).

```
given ( Val(housing_type, OtherCattle::Housing::Type) ) {
return Tech(er_no_solid_Slurry)                       when 'Tied_Housing_Slurry';
return Tech(er_no_solid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_solid_Slurry)                       when 'Loose_Housing_Slurry';
return Tech(er_no_solid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_solid_Solid)                        when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_n2o_nsolid Annual N2 emissions from cattles housing, yard and grazing (production).

```
given ( Val(housing_type, OtherCattle::Housing::Type) ) {
return Tech(er_n2o_solid_Slurry)                       when 'Tied_Housing_Slurry';
return Tech(er_n2o_solid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_solid_Slurry)                       when 'Loose_Housing_Slurry';
return Tech(er_n2o_solid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_solid_Solid)                        when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_n2_nliquid Annual N2 emissions from cattles housing, yard and grazing (production).

```
given ( Val(housing_type, OtherCattle::Housing::Type) ) {
return Tech(er_n2_liquid_Slurry)                       when 'Tied_Housing_Slurry';
return Tech(er_n2_liquid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_liquid_Slurry)                       when 'Loose_Housing_Slurry';
return Tech(er_n2_liquid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_liquid_Solid)                        when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_no_nliquid Annual N2 emissions from cattles housing, yard and grazing (production).

```
given ( Val(housing_type, OtherCattle::Housing::Type) ) {
return Tech(er_no_liquid_Slurry)                       when 'Tied_Housing_Slurry';
return Tech(er_no_liquid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_liquid_Slurry)                       when 'Loose_Housing_Slurry';
return Tech(er_no_liquid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_liquid_Solid)                        when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_n2o_nliquid Annual N2 emissions from cattles housing, yard and grazing (production).

```
given ( Val(housing_type, OtherCattle::Housing::Type) ) {
return Tech(er_n2o_liquid_Slurry)                       when 'Tied_Housing_Slurry';
return Tech(er_n2o_liquid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_liquid_Slurry)                       when 'Loose_Housing_Slurry';
return Tech(er_n2o_liquid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_liquid_Solid)                        when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

n2_nsolid Annual N2 emission from cattle housing, yard and grazing (production).

```
( Val(n_into_housing, OtherCattle::Housing) * (1-Val(share_liquid, OtherCattle::Housing::Type) )
) * Out(er_n2_nsolid);
```

no_nsolid Annual NO emission from cattle housing, yard and grazing (production).

```
( Val(n_into_housing, OtherCattle::Housing) * (1-Val(share_liquid, OtherCattle::Housing::Type) )
) * Out(er_no_nsolid);
```

n2o_nsolid Annual N2o emission from cattle housing, yard and grazing (production).

```
( Val(n_into_housing, OtherCattle::Housing) * (1-Val(share_liquid, OtherCattle::Housing::Type) )
) * Out(er_n2o_nsolid);
```

n2_nliquid Annual N2 emission from cattle housing, yard and grazing (production).

```
( Val(n_into_housing, OtherCattle::Housing) * Val(share_liquid, OtherCattle::Housing::Type)  +
Val(n_into_yard, OtherCattle::Yard)
) * Out(er_n2_nliquid);
```

no_nliquid Annual NO emission from cattle housing, yard and grazing (production).

```
( Val(n_into_housing, OtherCattle::Housing) * Val(share_liquid, OtherCattle::Housing::Type)  +
Val(n_into_yard, OtherCattle::Yard)
) * Out(er_no_nliquid);
```

n2o_nliquid Annual N2o emission from cattle housing, yard and grazing (production).

```
( Val(n_into_housing, OtherCattle::Housing) * Val(share_liquid, OtherCattle::Housing::Type)  +
Val(n_into_yard, OtherCattle::Yard)
) *  Out(er_n2o_nliquid);
```

1.84 Livestock::OtherCattle::Housing

This process calculates the NH3 emission in cattle housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emission, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.84.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.84.2 Outputs

n_into_housing Annual N flux into the house.

```
if( Val(n_excretion, Excretion) > (Val(n_into_grazing, Grazing) + Val(n_into_yard, Yard)) ) {
Val(n_excretion, Excretion)
- Val(n_into_grazing, Grazing)
- Val(n_into_yard, Yard);
} else { return 0; }
```

tan_into_housing Annual N flux as TAN into the house.

```
if( Val(tan_excretion, Excretion) > (Val(tan_into_grazing, Grazing) + Val(tan_into_yard, Yard)) ) {
Val(tan_excretion, Excretion)
- Val(tan_into_grazing, Grazing)
- Val(tan_into_yard, Yard);
} else { return 0; }
```

nh3_nhousing Annual NH3 emission from cattle housing systems per animal place.

```
Out(tan_into_housing)
* Val(c_grazing, Housing::KGrazing)
* Val(er_housing, Housing::Type)
* Val(c_area, Housing::KArea)
* (
(1 - Val(c_feeding_boxes, Housing::Type) )
* (1 - Val(c_UNECE, Housing::Floor) )
* (1 - Val(c_housing_floor, Housing::Floor) )
* (1 - Val(c_housing_climate, Housing::ClimateAir) )
* (1 - Val(c_housing_air, Housing::ClimateAir) )
)
* Val(c_free_factor_housing, Housing::CFreeFactor) ;
```

n_outhousing Annual N flux out of housing.

```
if( (Out(n_into_housing) - Out(nh3_nhousing)) > 0) {
Out(n_into_housing) - Out(nh3_nhousing);
} else { return 0; }
```

tan_outhousing Annual N flux as TAN out of housing.

```
if( (Out(tan_into_housing) - Out(nh3_nhousing)) > 0) {
Out(tan_into_housing) - Out(nh3_nhousing);
} else { return 0; }
```

share_tan_out Share of TAN for interface to storage, the same share is assumed for solid and liquid parts of output. Mineralization and immobilization are considerd in the storage module.

```
if(  Out(n_outhousing) != 0 ){
Out(tan_outhousing) / Out(n_outhousing);
}else{0};
```

n_outhousing_liquid Annual N flux out of housing, slurry or liquid fraction from cattle.

```
Out(n_outhousing) * Val(share_liquid, Housing::Type);
```

tan_outhousing_liquid Annual N flux as TAN out of housing, slurry or liquid fraction from cattle.

```
Out(n_outhousing) * Val(share_liquid, Housing::Type) * Out(share_tan_out);
```

n_outhousing_solid Annual N flux out of housing, solid manure fraction of N flux.

```
Out(n_outhousing) * ( 1 - Val(share_liquid, Housing::Type) );
```

tan_outhousing_solid Annual N flux as TAN out of housing, solid manure fraction of N flux.

```
Out(n_outhousing) * ( 1 - Val(share_liquid, Housing::Type) ) * Out(share_tan_out);
```

animalcategory Animal category

```
Val(animalcategory, Excretion);
```

1.85 Livestock::OtherCattle::Housing::KGrazing

This process calculates the correction factor for the non proportional change of the housing emission according to the grazing hours per day. Source is Phillips et al. (1998) and Gilhespy et al. (2006) empirical estimation was done by Thomas Kupper.

1.85.1 Outputs

c_grazing The correction factor for the reduction of the housing emission depending on the number of grazing hours per day and the grazing days per year.

```
my $k_grazing = Tech(k_grazing_a) * exp(Tech(k_grazing_b) * Val(grazing_hours, ..::GrazingInput));
# scale with ratio grazing_days per year
($k_grazing - 1) * Val(grazing_days, ..::GrazingInput) / 365 + 1;
```

1.86 Livestock::OtherCattle::Grazing

This process calculates the annual NH3 emission from grazing cattle based on the total N excreted on the pastures. The N excreted is calculated according to the time the animals spend on the pasture. It assumes that the excretions on the pasture are proportional to the time the animals are grazed.

1.86.1 References:

Bussink DW 1992. Ammonia volatilization from grassland receiving nitrogen fertilizer and rotationally grazed by dairy cattle. Fertilizer Research 33:257-265.

Bussink DW 1994. Relationship between ammonia volatilization and nitrogen fertilizer application rate, intake and excretion of herbage nitrogen by cattle on grazed swards. Fertilizer Research 38:111-121.

Jarvis SC, Hatch DJ, Lockyer DR 1989. Ammonia fluxes from grazed grassland: annual losses from cattle production systems and their relation to nitrogen inputs. Journal of Agricultural Science 113:99-108.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Ross CA, Jarvis SC 2001. Measurement of emission and deposition pattern of ammonia from urine in grass swards. Atmospheric environment 35:867-875.

1.86.2 Outputs

n_into_grazing Annual N excretion during grazing for a defined cattle category.

```
Val(n_excretion, Excretion) *
Val(grazing_hours, GrazingInput) / 24 *
(
(Val(grazing_days, GrazingInput) - Val(days_with_grazing_and_yard, Yard)) / 365 +
(1 - Val(share_excretion_with_grazing, Yard)) *
Val(days_with_grazing_and_yard, Yard) / 365
);
```

tan_into_grazing Annual soluble N (TAN) excretion during grazing for a defined cattle category.

```
Val(tan_excretion, Excretion) *
Val(grazing_hours, GrazingInput) / 24 *
(
(Val(grazing_days, GrazingInput) - Val(days_with_grazing_and_yard, Yard)) / 365 +
(1 - Val(share_excretion_with_grazing, Yard)) *
Val(days_with_grazing_and_yard, Yard) / 365
);
```

ef_nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
Tech(er_cattle_grazing);
```

nh3_ngrazing Annual NH3 emission from all grazing cattle.

```
Out(tan_into_grazing)  * Out(ef_nh3_ngrazing);
```

n2_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2_cattle_grazing);
```

no_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_no_cattle_grazing);
```

n2o_ngrazing Annual total N2O emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2o_cattle_grazing);
```

n_remain_grazing Annual N input on pastures.

```
Out(n_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

tan_remain_grazing Annual N input on pastures.

```
Out(tan_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

1.87 Livestock::OtherCattle::Yard

1.87.1 Inputs

yard_days Access to exercise yards days per year.

exercise_yard Exercise yard: not available, available: roughage is not supplied in the exercise yard, available: roughage is partly supplied in the exercise yard, available: roughage is exclusively supplied in the exercise yard.

floor_properties_exercise_yard_SHL Floor properties(solid floor, unpaved floor, perforated floor, paddock or pasture used as exercise yard).

floor_properties_exercise_yard_LU Type of exercise yard.

free_correction_factor Factor to define free ?

1.87.2 Outputs

days_with_grazing_and_yard Number of Days whith access to yard and pasture

```
if( (Val(grazing_days, GrazingInput) + In(yard_days)) > 365 ){
my $yardDays = In(yard_days);
my $grazingDays = Val(grazing_days, GrazingInput);
writeLog({en=>"The sum of days with access to the exercise yard and of grazing days per year for other cattle
de=>"Die Summe der Weide- und Laufhoftage für übriges Rindvieh ist grösser als 365!\n",
fr=>"La somme des jours de pâturage et des jours avec accès au parcours extérieur pour "
. "la catégorie autres bovons est supérieure à 365 !\n",
});
return ( Val(grazing_days, GrazingInput) + In(yard_days) - 365) ;
} else{
return (0);
};
```

c_free_factor_yard Free correction factor of the Emission rate for the Yard.

```
if(In(free_correction_factor)//0 != 0) {
writeLog({en=>"You have entered an additional emission mitigation measure for an exercise yard of the categor
. In(free_correction_factor)
```

```
. "\%!\n",
de=>"Sie haben eine zusätzliche emissionsmindernde  Massnahme für einen Laufhof der Kategorie übriges Rindvie
. In(free_correction_factor)
. "\% eingegeben!\n",
fr=>"Vous avez introduit une mesure supplémentaire limitant les émissions provenant d'un parcours extérieur "
. "pour autres bovins de ". In(free_correction_factor) . "\%.\n"
});
return 1 - In(free_correction_factor)/100;

} else {
return 1;
}
```

c_floor_properties_exercise_yard_SHL Correction factor for the emission due to the use of the floor properties in housing systems.

```
given ( In(floor_properties_exercise_yard_SHL) ) {
when 'solid_floor' {
1 - Tech(red_floor_properties_solid_floor);
}
when  'unpaved_floor' {
1 - Tech(red_floor_properties_unpaved_floor);
}
when  'perforated_floor' {
1 - Tech(red_floor_properties_perforated_floor);
}
when  'paddock_or_pasture_used_as_exercise_yard' {
1 - Tech(red_floor_properties_paddock_or_pasture_used_as_exercise_yard);
}
default {
1;
}
}
```

c_floor_properties_exercise_yard_LU Correction factor for the emission due to the use of the floor properties in housing systems.

```
return 1 unless defined In(floor_properties_exercise_yard_LU);
given ( In(floor_properties_exercise_yard_LU) ) {
when 'solid_floor' {
1 - Tech(red_floor_properties_solid_floor);
}
when 'perforated_floor' {
1 - Tech(red_floor_properties_perforated_floor);
}
when 'sloped_concrete_floor_with_central_drain_and_stationary_scraper' {
1 - Tech(red_sloped_concrete_floor_with_central_drain_and_stationary_scraper);
}
when 'sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading' {
1 - Tech(red_sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading);
}
when 'sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading_plus_shower'
1 - Tech(red_sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading_plus_
}
default {
1;
}
}
```

share_excretion Share of excretion on the yard according the stay on yard. If the basic feeding is on the yard the share_basicfeeding is assumed. Otherwise the share is selected according to the access.

```
given ( In(exercise_yard) ) {
when 'available_roughage_is_not_supplied_in_the_exercise_yard' {
Tech(share_available_roughage_is_not_supplied_in_the_exercise_yard);
}
when 'available_roughage_is_partly_supplied_in_the_exercise_yard' {
```

```
Tech(share_available_roughage_is_partly_supplied_in_the_exercise_yard);
}
when 'available_roughage_is_exclusively_supplied_in_the_exercise_yard' {
Tech(share_available_roughage_is_exclusively_supplied_in_the_exercise_yard);
}
when 'not_available' {
0;
}
}
```

share_excretion_with_grazing Share of excretion on the yard according the stay on yard with parallel acess to Pasture. If the basic feeding is on the yard the share_basicfeeding is assumed. Otherwise the share is selected according to the access.

```
if( (Out(days_with_grazing_and_yard) > 0) and (Out(share_excretion) > Tech(share_available_roughage_is_partly
writeLog({
en => "The category \"roughage is exclusively supplied in the exercise yard\" is not compatible with the numb
de => "Verabreichung von Grundfutter ausschliesslich auf dem Laufhof ist nicht möglich mit den eingegebenen W
fr => "La distribution de fourrage exclusivement dans le parcours extérieur "
. "pour les vâches laitières n'est pas possible pendant les jours de pâturage !\n",
});
return Tech(share_available_roughage_is_partly_supplied_in_the_exercise_yard);
} else {
return Out(share_excretion);
}
```

n_into_yard Annual N excretion on yard for a defined animal category.

```
Val(n_excretion, Excretion) *
(
Out(share_excretion) *
(In(yard_days) - Out(days_with_grazing_and_yard)) / 365 +
Out(share_excretion_with_grazing) *
Out(days_with_grazing_and_yard) / 365
);
```

tan_into_yard Annual soluble N excretion on yard for a defined animal category.

```
Val(tan_excretion,Excretion) *
(
Out(share_excretion) *
(In(yard_days) - Out(days_with_grazing_and_yard)) / 365 +
Out(share_excretion_with_grazing) *
Out(days_with_grazing_and_yard) / 365
);
```

ef_nh3_nyard Annual NH3 emission from yard.

```
Tech(er_yard) *
Out(c_floor_properties_exercise_yard_SHL) *
Out(c_floor_properties_exercise_yard_LU) *
Out(c_free_factor_yard);
```

nh3_nyard Annual NH3 emission from yard.

```
Out(tan_into_yard) * Out(ef_nh3_nyard);
```

n_outyard_liquid Annual N flux from liquid part out of yard.

```
Out(n_into_yard) - Out(nh3_nyard);
```

tan_outyard_liquid Annual N flux as TAN from liquid part out of yard into storage.

```
Out(tan_into_yard) - Out(nh3_nyard);
```

n_outyard_solid Annual N flux from solid part out of yard.

```
0;
```

tan_outyard_solid Annual N flux as TAN from solid part out of yard into storage.

```
0;
```

1.88 Livestock::OtherCattle::GrazingInput

Input parameters for Grazing, as it is used also in Yard in separete Module.

1.88.1 Inputs

grazing_days Average grazing days per year.

grazing_hours Average grazing hours per day.

1.88.2 Outputs

grazing_hours Grazing hours per day.

```
In(grazing_hours);
```

grazing_days Grazing days per year.

```
In(grazing_days);
```

1.89 Livestock::OtherCattle::Housing::CFreeFactor

1.89.1 Inputs

free_correction_factor Factor to define free.

1.89.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a housing of the category other cattle
. In(free_correction_factor) . "\%!\n",
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Stall der Kategorie Übriges Rindvieh
. In(free_correction_factor) . "\% eingegeben!\n" ,
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions "
. "provenant d'une stabulation pour autres bovins de " . In(free_correction_factor) . "\%.\n"
});
return 1 - In(free_correction_factor) / 100;
}
else {
return 1;
}
```

1.90 Livestock::OtherCattle::Housing::ClimateAir

This submodul calculates the annual NH3 reduction due to a grooved floor in housing systems according to the UNECE guideline 2007.

1.90.1 References

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.90.2 Inputs

mitigation_options_for_housing_systems_for_other_cattle_climate Mitigation option housing climate

mitigation_options_for_housing_systems_for_other_cattle_air Mitigation option ventilation

1.90.3 Outputs

c_housing_climate Reduction factor for the emission due to the use of housing system adaptations.

```
return 0 unless defined In(mitigation_options_for_housing_systems_for_other_cattle_climate);
if (In(mitigation_options_for_housing_systems_for_other_cattle_climate) eq 'roof_insulation'){
return( Tech(red_roof_insulation));
} elsif (In(mitigation_options_for_housing_systems_for_other_cattle_climate) eq 'water_spraying'){
return( Tech(red_water_spraying));
} elsif (In(mitigation_options_for_housing_systems_for_other_cattle_climate) eq 'roof_insulation_inclusive_wa
return( Tech(red_roof_insulation_inclusive_water_spraying));
} elsif (In(mitigation_options_for_housing_systems_for_other_cattle_climate) eq 'sprinkling_onto_the_roof'){
return( Tech(red_sprinkling_onto_the_roof));
} elsif (In(mitigation_options_for_housing_systems_for_other_cattle_climate) eq 'sprinkling_onto_the_roof_inc
return( Tech(red_sprinkling_onto_the_roof_inclusive_water_spraying));
} else { return 0;};
```

c_housing_air Reduction factor for the emission due to the use of housing system adaptations.

```
return 0 unless defined In(mitigation_options_for_housing_systems_for_other_cattle_air);
if (In(mitigation_options_for_housing_systems_for_other_cattle_air)  eq 'well_adapted_aeration'){
return( Tech(red_well_adapted_aeration));
} else { return 0;};
```

1.91 Livestock::OtherCattle::Housing::Floor

This submodul calculates the annual NH3 reduction due to a grooved floor in cattle housing systems according to the UNECE guideline 2007.

1.91.1 References

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.91.2 Inputs

UNECE_category_1_mitigation_options_for_housing_systems_for_other_cattle UNECE category 1 mitigation options for loose housing systems for other cattle (none, toothed scrapper running over a grooved floor).

mitigation_options_for_housing_systems_for_other_cattle_floor Mitigation options floor for housing systems for other cattle.

1.91.3 Outputs

c_UNECE Reduction factor for the emission due to the use of a grooved floor in housing systems.

```
return 0 unless defined In(UNECE_category_1_mitigation_options_for_housing_systems_for_other_cattle);
if (In(UNECE_category_1_mitigation_options_for_housing_systems_for_other_cattle) eq 'raised_feeding_stands')
return( Tech(red_raised_feeding_stands) );
}
elsif (In(UNECE_category_1_mitigation_options_for_housing_systems_for_other_cattle) eq 'floor_with_cross_slop
return( Tech(red_floor_with_cross_slope_and_collection_gutter) );
}
elsif (In(UNECE_category_1_mitigation_options_for_housing_systems_for_other_cattle) eq 'floor_with_cross_slop
return( Tech(red_floor_with_cross_slope_and_collection_gutter_and_raised_feeding_stands) );
}
elsif (In(UNECE_category_1_mitigation_options_for_housing_systems_for_other_cattle) eq 'toothed_scrapper_runn
return( Tech(red_toothed_scrapper_running_over_a_grooved_floor) );
}
else {
return 0;
}



#if (In(UNECE_category_1_mitigation_options_for_housing_systems_for_other_cattle) eq 'toothed_scrapper_runnin
#       return( Tech(red_UNECE));
#       } else { return 0;};
```

c_housing_floor Reduction factor for the emission due to the use of housing system adaptations.

```
return 0 unless defined In(mitigation_options_for_housing_systems_for_other_cattle_floor);
if (In(mitigation_options_for_housing_systems_for_other_cattle_floor) eq 'toothed_scrapper_running_over_a_gro
return( Tech(red_toothed_scrapper_running_over_a_grooved_floor));
} elsif (In(mitigation_options_for_housing_systems_for_other_cattle_floor) eq 'sloped_concrete_floor_with_dra
return( Tech(red_sloped_concrete_floor_with_drain_and_stationary_scarp_cleaning_every_two_hours));
} elsif (In(mitigation_options_for_housing_systems_for_other_cattle_floor) eq 'system_Comfort_Slat_Mat_with_M
return( Tech(red_system_Comfort_Slat_Mat_with_Mat_and_Valve));
} else { return 0;};
```

1.92 Livestock::OtherCattle::Housing::KArea

It is assumed that additional areas in stable, as example if the stabel is built for more animal, are fully used. So the emissions increase 5
the double of the regularized minimal area.

TODO (All): Provide Articles which confirm/justify the assumption. As decided on workshop from 2.11.2007.

1.92.1 Inputs

dimensioning_barn Number of available animal places.

1.92.2 Outputs

dimensioning_barn Number of anymal places (needed in other modules).

```
In(dimensioning_barn);
```

area_increase Factor on what barn size does increase the regularized minimal, limited to 0.5

```
if ( (Val(animals, ..::Excretion) < In(dimensioning_barn) ) and (Val(animals, ..::Excretion) != 0) ) {
if ( In(dimensioning_barn) >= (Val(animals, ..::Excretion) * 1.5) ) {
0.5;
}
else {
```

```
In(dimensioning_barn) / Val(animals, ..::Excretion) - 1;
}
}
else {
0.0;
}
```

c_area Correction factor for area per animal.

```
1 + (Out(area_increase) * Val(k_area, Type));
```

1.93 Livestock::OtherCattle::Housing::Type

This process selects the correction factor for the specific housing types for cattle. Among these correction factors are the emission rate for the housing type, the correction factor for the housing type area, the correction factor for the area per animal, the liquid and solid share for the housing type as well as the factor on what barn size increases the regularized minimal (limited to 0.5).

1.93.1 Inputs

housing_type Type of housing.

feeding_boxes Are feeding boxes available?

1.93.2 Outputs

er_housing Emission rate for the housing type.

```
if ( In(housing_type) eq 'Tied_Housing_Slurry' ) {
return Val(er_housing,Type::Tied_Housing_Slurry);
}
elsif ( In(housing_type) eq 'Tied_Housing_Slurry_Plus_Solid_Manure')  {
return Val(er_housing,Type::Tied_Housing_Slurry_Plus_Solid_Manure);
}
elsif ( In(housing_type) eq 'Loose_Housing_Slurry') {
return Val(er_housing,Type::Loose_Housing_Slurry);
}
elsif ( In(housing_type) eq 'Loose_Housing_Slurry_Plus_Solid_Manure') {
return Val(er_housing,Type::Loose_Housing_Slurry_Plus_Solid_Manure);
}
elsif ( In(housing_type) eq 'Loose_Housing_Deep_Litter') {
return Val(er_housing,Type::Loose_Housing_Deep_Litter);
}
```

share_liquid Liquid share for the housing type.

```
if ( In(housing_type) eq 'Tied_Housing_Slurry' ) {
return Val(share_liquid,Type::Tied_Housing_Slurry);
}
elsif ( In(housing_type) eq 'Tied_Housing_Slurry_Plus_Solid_Manure')  {
return Val(share_liquid,Type::Tied_Housing_Slurry_Plus_Solid_Manure);
}
elsif ( In(housing_type) eq 'Loose_Housing_Slurry') {
return Val(share_liquid,Type::Loose_Housing_Slurry);
}
elsif ( In(housing_type) eq 'Loose_Housing_Slurry_Plus_Solid_Manure') {
return Val(share_liquid,Type::Loose_Housing_Slurry_Plus_Solid_Manure);
}
elsif ( In(housing_type) eq 'Loose_Housing_Deep_Litter') {
```

```
return Val(share_liquid,Type::Loose_Housing_Deep_Litter);
}
```

share_solid Solid share for the housing type.

```
if ( In(housing_type) eq ’Tied_Housing_Slurry’ ) {
return Val(share_solid,Type::Tied_Housing_Slurry);
}
elsif ( In(housing_type) eq ’Tied_Housing_Slurry_Plus_Solid_Manure’)  {
return Val(share_solid,Type::Tied_Housing_Slurry_Plus_Solid_Manure);
}
elsif ( In(housing_type) eq ’Loose_Housing_Slurry’) {
return Val(share_solid,Type::Loose_Housing_Slurry);
}
elsif ( In(housing_type) eq ’Loose_Housing_Slurry_Plus_Solid_Manure’) {
return Val(share_solid,Type::Loose_Housing_Slurry_Plus_Solid_Manure);
}
elsif ( In(housing_type) eq ’Loose_Housing_Deep_Litter’) {
return Val(share_solid,Type::Loose_Housing_Deep_Litter);
}
```

k_area Correction factor for the housing type area.

```
given ( In(housing_type) ) {
when ’Tied_Housing_Slurry’ {
return Val(k_area, Type::Tied_Housing_Slurry);
}
when ’Tied_Housing_Slurry_Plus_Solid_Manure’ {
return Val(k_area, Type::Tied_Housing_Slurry_Plus_Solid_Manure);
}
when ’Loose_Housing_Slurry’ {
return Val(k_area, Type::Loose_Housing_Slurry);
}
when ’Loose_Housing_Slurry_Plus_Solid_Manure’ {
return Val(k_area, Type::Loose_Housing_Slurry_Plus_Solid_Manure);
}
when ’Loose_Housing_Deep_Litter’ {
return Val(k_area, Type::Loose_Housing_Deep_Litter);
}
}
```

housing_type Housing type (needed in other modules).

```
In(housing_type);
```

c_feeding_boxes Are feeding boxes available?

```
if( defined In(feeding_boxes)  and In(feeding_boxes) eq ’yes’ ){
if( (In(housing_type) eq ’Tied_Housing_Slurry’) or (In(housing_type) eq ’Tied_Housing_Slurry_Plus_Solid_Manur
writeLog({en=>"Feeding are applicable for loose housing systems only!",
de=>"Fütterungsbuchten sind nur für Laufställe anwendbar!",
fr=>"Feeding boxes sont éligible seulement pour les stabulations libre!"});
return 0;
}
else{
return( Tech(red_feeding_boxes) );
}
}
else {
return 0;
};
```

1.94 Livestock::OtherCattle::Housing::Type::Loose_Housing_Deep_Litter

This process describes the correction factors for the loose housing deep litter system such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.94.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. UNECE executive body for the convention on long-range transboundary air pollution, Working Group on Strategies and Review, EB.AIR/WH.5/2007/13/ 16 July 2007, Genf. 35p.

1.94.2 Inputs

ignore Just a work around for modules without input parameters.

1.94.3 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

share_solid Share of solid manure of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.95 Livestock::OtherCattle::Housing::Type::Loose_Housing_Slurry_Plus_Solid_Manure

This process describes the correction factors for the loose housing liquid solid system such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.95.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. UNECE executive body for the convention on long-range transboundary air pollution, Working Group on Strategies and Review, EB.AIR/WH.5/2007/13/ 16 July 2007, Genf. 35p.

1.95.2 Inputs

ignore Just a work around for modules without input parameters.

1.95.3 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er),
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid),
```

share_solid Share of solid manure of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.96 Livestock::OtherCattle::Housing::Type::Loose_Housing_Slurry

This process describes the correction factors for the loose housing slurry system such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.96.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. UNECE executive body for the convention on long-range transboundary air pollution, Working Group on Strategies and Review, EB.AIR/WH.5/2007/13/ 16 July 2007, Genf. 35p.

1.96.2 Inputs

ignore Just a work around for modules without input parameters.

1.96.3 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

share_solid Share of solid manure of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.97 Livestock::OtherCattle::Housing::Type::Tied_Housing_Slurry_Plus_Solid_Manure

This process describes the correction factors for the tied housing liquid solid system such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.97.1 Inputs

ignore Just a work around for modules without input parameters.

1.97.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

share_solid Share of solid manure of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.98 Livestock::OtherCattle::Housing::Type::Tied_Housing_Slurry

This process describes the correction factors for the tied housing slurry system such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.98.1 Inputs

ignore Just a work around for modules without input parameters.

1.98.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

share_solid Share of solid manure of Ntot for selected housing type.

```
1-Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.99 Livestock::OtherCattle::Excretion

This process calculates the annual N excretion of a number of cattle as a function of the supplied feed ration. The standard N excretion was taken from the official Swiss fertilizer guidelines. The Nsol content of the excreta is based on e.g. Peterson et al. (1998) or Burgos et al. (2005). These values were compiled on the basis of official feeding recommendations (Flisch et al. 2009) by H. Menzi. A detailed documentation will be prepared in the framework of the new revision of the document in the course of summer 2007.

1.99.1 References:

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.99.2 Inputs

animalcategory Animal category (suckling cows, 1 year old heifers, 2 years old heifers, 3 years old heifers, fattening calves, calves of suckling cows, beef cattle).

animals Number of animals for the selected type in barn.

inp_n_excretion Annual standard N excretion for a dairy cow

1.99.3 Outputs

animals Number of animals for the selected cattle category in barn.

```
In(animals);
```

animalcategory Animal category

```
In(animalcategory);
```

n_excretion_animal Annual total N excreted per animal.

```
my $cat = Out(animalcategory);
my $key = 'standard_N_excretion_' . $cat;
my $exc = $TE->{$key};
$exc = In(inp_n_excretion) unless lc In(inp_n_excretion) eq 'standard';
if ( ($exc < 0.25 * $TE->{$key}) or ($exc > 1.75 * $TE->{$key})) {
writeLog({
en => "FIX: Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
de => "Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
fr => "FIX: Die eingegebene N-Ausscheidung für $cat weicht mehr als 75% vom Standard ab.",
});
}
return $exc;
```

n_excretion Annual total N excreted by an animalgroup of selected cattle category.

```
Out(n_excretion_animal) * Out(animals);
```

tan_excretion Annual soluble N excreted by an animalgroup of selected cattle category.

```
$TE->{"share_Nsol_" . Out(animalcategory)} * Out(n_excretion);
```

1.100 Livestock::DairyCow

XXX

1.100.1 Outputs

dimensioning_check Check if number of animals <= number of animal places.

```
if ( Val(dimensioning_barn, DairyCow::Housing::KArea) < Val(animals, DairyCow::Excretion) ) {
writeLog({
en => "The number of available animal places must be equal or higher than the number of animals.",
de => "Anzahl Tierplätze muss grösser oder gleich der Anzahl der Tiere sein.",
fr => "Le nombre de places dans l'étable doit être au moins égal au nombre d'animaux."
});
return "not ok";
}
else {
return "ok";
}
```

er_n2_nsolid Annual N2 emissions from dairy cows housing, yard and grazing (production).

```
given ( Val(housing_type, DairyCow::Housing::Type) ) {
return Tech(er_n2_solid_Slurry)                     when 'Tied_Housing_Slurry';
return Tech(er_n2_solid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_solid_Slurry)                     when 'Loose_Housing_Slurry';
return Tech(er_n2_solid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_solid_Solid)                      when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_no_nsolid Annual N2 emissions from dairy cows housing, yard and grazing (production).

```
given ( Val(housing_type, DairyCow::Housing::Type) ) {
return Tech(er_no_solid_Slurry)                     when 'Tied_Housing_Slurry';
return Tech(er_no_solid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_solid_Slurry)                     when 'Loose_Housing_Slurry';
return Tech(er_no_solid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_solid_Solid)                      when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_n2o_nsolid Annual N2 emissions from dairy cows housing, yard and grazing (production).

```
given ( Val(housing_type, DairyCow::Housing::Type) ) {
return Tech(er_n2o_solid_Slurry)                     when 'Tied_Housing_Slurry';
return Tech(er_n2o_solid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_solid_Slurry)                     when 'Loose_Housing_Slurry';
return Tech(er_n2o_solid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_solid_Solid)                      when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_n2_nliquid Annual N2 emissions from dairy cows housing, yard and grazing (production).

```
given ( Val(housing_type, DairyCow::Housing::Type) ) {
return Tech(er_n2_liquid_Slurry)                     when 'Tied_Housing_Slurry';
return Tech(er_n2_liquid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_liquid_Slurry)                     when 'Loose_Housing_Slurry';
return Tech(er_n2_liquid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2_liquid_Solid)                      when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_no_nliquid Annual N2 emissions from dairy cows housing, yard and grazing (production).

```
given ( Val(housing_type, DairyCow::Housing::Type) ) {
return Tech(er_no_liquid_Slurry)                     when 'Tied_Housing_Slurry';
return Tech(er_no_liquid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_liquid_Slurry)                     when 'Loose_Housing_Slurry';
return Tech(er_no_liquid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_no_liquid_Solid)                      when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

er_n2o_nliquid Annual N2 emissions from dairy cows housing, yard and grazing (production).

```
given ( Val(housing_type, DairyCow::Housing::Type) ) {
return Tech(er_n2o_liquid_Slurry)                     when 'Tied_Housing_Slurry';
return Tech(er_n2o_liquid_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_liquid_Slurry)                     when 'Loose_Housing_Slurry';
return Tech(er_n2o_liquid_Slurry_Plus_Solid_Manure)  when 'Loose_Housing_Slurry_Plus_Solid_Manure';
return Tech(er_n2o_liquid_Solid)                      when 'Loose_Housing_Deep_Litter';
default { return 0; }
};
```

n2_nsolid FIX: Annual N2 emissions from dairy cows housing, yard and storage.

```
( Val(n_into_housing, DairyCow::Housing) * (1 - Val(share_liquid, DairyCow::Housing::Type) )
) * Out(er_n2_nsolid);
```

no_nsolid FIX: Annual NO emissions from dairy cows housing, yard and grazing (production).

```
( Val(n_into_housing, DairyCow::Housing) * (1 - Val(share_liquid, DairyCow::Housing::Type) )
) * Out(er_no_nsolid);
```

n2o_nsolid FIX: Annual N2O emissions from dairy cows housing, yard and grazing (production).

```
( Val(n_into_housing, DairyCow::Housing) * (1 - Val(share_liquid, DairyCow::Housing::Type) )
) * Out(er_n2o_nsolid);
```

n2_nliquid FIX: Annual N2 emissions from dairy cows housing, yard and grazing (production).

```
( Val(n_into_housing, DairyCow::Housing) * Val(share_liquid, DairyCow::Housing::Type)  +
Val(n_into_yard, DairyCow::Yard)
) * Out(er_n2_nliquid);
```

no_nliquid FIX: Annual NO emissions from dairy cows housing, yard and grazing (production).

```
( Val(n_into_housing, DairyCow::Housing) * Val(share_liquid, DairyCow::Housing::Type)  +
Val(n_into_yard, DairyCow::Yard)
) * Out(er_no_nliquid);
```

n2o_nliquid FIX: Annual N2O emissions from dairy cows housing, yard and grazing (production).

```
( Val(n_into_housing, DairyCow::Housing) * Val(share_liquid, DairyCow::Housing::Type)  +
Val(n_into_yard, DairyCow::Yard)
) *  Out(er_n2o_nliquid);
```

1.101 Livestock::DairyCow::Housing

This process calculates the NH3 emission in dairy cow housing depending on the N excretion and the housing systems. The NH3 emission is assumed to be proportional to the total N excretion of the animals because the contaminated surfaces will primarily drive emission, which will remain active even when animals are on the pasture or exercise yard.

A proportional correction is applied according to the area per animal.

A correction is applied if the part of the day grazing is above a defined grazing hours per days.

1.101.1 Definition of System boundaries for the housing Process:

For housing systems with integrated exercise yards, it is difficult to distinguish between emissions from housing, exercise yard and storage. For these housingtypes the emission may be only validated for the sum of housing, exercise yard and storage emission.

1.101.2 Outputs

n_into_housing Annual N flux into the house.

```
if( Val(n_excretion, Excretion) > (Val(n_into_grazing, Grazing) + Val(n_into_yard, Yard)) ) {
Val(n_excretion, Excretion) -
Val(n_into_grazing, Grazing) -
Val(n_into_yard, Yard);
} else {
0;
}
```

tan_into_housing Annual N flux as TAN into the house.

```
if( Val(tan_excretion, Excretion) > (Val(tan_into_grazing, Grazing) + Val(tan_into_yard, Yard)) ) {
Val(tan_excretion, Excretion) -
Val(tan_into_grazing, Grazing) -
Val(tan_into_yard, Yard);
} else {
0;
}
```

ef_nh3_nhousing NH3 emission factor for dairy cow housing systems.

```
my $c_housing = Val(c_grazing, Housing::KGrazing) *
Val(c_area, Housing::KArea) *
Val(er_housing, Housing::Type) *
Val(c_feeding_boxes, Housing::Type) *
Val(c_UNECE, Housing::Floor) *
Val(c_housing_floor, Housing::Floor) *
Val(c_free_factor_housing, Housing::CFreeFactor);
#FIXME: Check: Is it in any way even possible that $c_housing > 1???
$c_housing = 1 unless $c_housing < 1;
return $c_housing;
```

nh3_nhousing Annual NH3 emission from dairy cow housing systems per animal place.

```
Out(tan_into_housing) * Out(ef_nh3_nhousing);
```

n_outhousing Annual N flux out of the housing.

```
Out(n_into_housing) - Out(nh3_nhousing);
```

tan_outhousing Annual N flux as TAN out of the housing.

```
Out(tan_into_housing) - Out(nh3_nhousing);
```

n_outhousing_liquid Annual N flux out of housing, slurry or liquid fraction of manure from dairy cows.

```
Out(n_outhousing) * Val(share_liquid, Housing::Type);
```

tan_outhousing_liquid Annual N flux as TAN out of housing, slurry or liquid fraction of manure from dairy cows.

```
Out(tan_outhousing) * Val(share_liquid, Housing::Type);
```

n_outhousing_solid Annual N flux out of housing from solid fraction of manure.

```
Out(n_outhousing) - Out(n_outhousing_liquid);
```

tan_outhousing_solid Annual N flux as TAN out of housing from solid fraction of manure.

```
Out(tan_outhousing) - Out(tan_outhousing_liquid);
```

1.102 Livestock::DairyCow::Housing::KGrazing

This process calculates the correction factor for the non proportional change of the housing emission according to the grazing hours per day. Source is Phillips et al. (1998) and Gilhespy et al. (2006) empirical estimation was done by Thomas Kupper.

1.102.1 Outputs

c_grazing The correction factor for the reduction of the housing emission depending on the number of grazing hours per day and the grazing days per year.

```
my $k_grazing = Tech(k_grazing_a) * exp(Tech(k_grazing_b) * Val(grazing_hours, ..::GrazingInput));
# scale with ratio grazing_days per year
($k_grazing - 1) * Val(grazing_days, ..::GrazingInput) / 365 + 1;
```

1.103 Livestock::DairyCow::Grazing

This process calculates the annual NH3 emission from grazing dairy cows based on the total N excreted on the pastures. The N excreted is calculated according to the time the animals spend on the pasture. It assumes that the excretions on the pasture are proportional to the time the animals are grazed.

1.103.1 References:

Bussink DW 1992. Ammonia volatilization from grassland receiving nitrogen fertilizer and rotationally grazed by dairy cattle. Fertilizer Research 33:257-265.

Bussink DW 1994. Relationship between ammonia volatilization and nitrogen fertilizer application rate, intake and excretion of herbage nitrogen by cattle on grazed swards. Fertilizer Research 38:111-121.

Jarvis SC, Hatch DJ, Lockyer DR 1989. Ammonia fluxes from grazed grassland: annual losses from cattle production systems and their relation to nitrogen inputs. Journal of Agricultural Science 113:99-108.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

Ross CA, Jarvis SC 2001. Measurement of emission and deposition pattern of ammonia from urine in grass swards. Atmospheric environment 35:867-875.

1.103.2 Outputs

n_into_grazing Annual total N excretion during grazing for dairy cows.

```
Val(n_excretion, Excretion) *
Val(grazing_hours, GrazingInput) / 24 *
(
(Val(grazing_days, GrazingInput) - Val(days_with_grazing_and_yard, Yard)) / 365 +
(1 - Val(share_excretion_with_grazing, Yard)) *
Val(days_with_grazing_and_yard, Yard) / 365
);
```

tan_into_grazing Annual soluble N (TAN) excretion during grazing for dairy cows.

```
Val(tan_excretion, Excretion) *
Val(grazing_hours, GrazingInput) / 24 *
(
(Val(grazing_days, GrazingInput) - Val(days_with_grazing_and_yard, Yard)) / 365 +
(1 - Val(share_excretion_with_grazing, Yard)) *
Val(days_with_grazing_and_yard, Yard) / 365
);
```

ef_nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
Tech(er_dairycow_grazing);
```

nh3_ngrazing Annual total NH3 emission from all grazing dairy cows.

```
Out(tan_into_grazing) * Out(ef_nh3_ngrazing);
```

n2_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2_dairycow_grazing);
```

no_ngrazing Annual total N2 emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_no_dairycow_grazing);
```

n2o_ngrazing Annual total N2O emission from all grazing dairy cows.

```
Out(n_into_grazing) * Tech(er_n2o_dairycow_grazing);
```

n_remain_grazing Annual N input on pasture.

```
Out(n_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

tan_remain_grazing Annual N input on pasture.

```
Out(tan_into_grazing) -
Out(nh3_ngrazing) -
Out(n2_ngrazing) -
Out(no_ngrazing) -
Out(n2o_ngrazing);
```

1.104 Livestock::DairyCow::Yard

1.104.1 Inputs

yard_days Access to exercise yard in days per year.

exercise_yard Exercise yard: not available, available: roughage is not supplied in the exercise yard, available: roughage is partly supplied in the exercise yard, available: roughage is exclusively supplied in the exercise yard.

floor_properties_exercise_yard_SHL Floor properties(solid floor, unpaved floor, perforated floor, paddock or pasture used as exercise yard).

floor_properties_exercise_yard_LU Type of exercise yard.

free_correction_factor Factor to define free ?

1.104.2 Outputs

days_with_grazing_and_yard Number of Days whith access to yard and pasture

```
if( (Val(grazing_days, GrazingInput) + In(yard_days)) > 365 ){
writeLog({
en => "The sum of days with access to the exercise yard and of grazing "
. "days per year for dairy cows is higher than 365!\n",
de => "Die Summe der Weide- und Laufhoftage für Milchkühe ist grösser als 365!\n",
fr => "La somme des jours de pâturage et des jours avec accès au parcours extérieur "
. "pour les vâches laitières est supérieure à 365 !\n",
});
return Val(grazing_days, GrazingInput) + In(yard_days) - 365;
```

```
} else {
return 0;
}
```

c_free_factor_yard Free reduction of the Emission rate for the Yard.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a dairy cows exercise yard of "
. In(free_correction_factor) . "\%!\n",
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Laufhof der Milchkühe von "
. In(free_correction_factor) . "\% eingegeben!\n",
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions provenant d’un parcours "
. "extérieur pour vâches laitières de " . In(free_correction_factor) . "\% !\n"
});
return 1 - In(free_correction_factor) / 100;
} else {
return 1;
}
```

c_floor_properties_exercise_yard_SHL Correction factor for the emission due to the use of the floor properties in housing systems.

```
given ( In(floor_properties_exercise_yard_SHL) ) {
when ’solid_floor’ {
1 - Tech(red_floor_properties_solid_floor);
}
when ’unpaved_floor’ {
1 - Tech(red_floor_properties_unpaved_floor);
}
when ’perforated_floor’ {
1 - Tech(red_floor_properties_perforated_floor);
}
when ’paddock_or_pasture_used_as_exercise_yard’ {
1 - Tech(red_floor_properties_paddock_or_pasture_used_as_exercise_yard);
}
default {
1;
}
}
```

c_floor_properties_exercise_yard_LU Correction factor for the emission due to the use of the floor properties in housing systems.

```
return 1 unless defined In(floor_properties_exercise_yard_LU);
given ( In(floor_properties_exercise_yard_LU) ) {
when ’solid_floor’ {
1 - Tech(red_floor_properties_solid_floor);
}
when ’perforated_floor’ {
1 - Tech(red_floor_properties_perforated_floor);
}
when ’sloped_concrete_floor_with_central_drain_and_stationary_scraper’ {
1 - Tech(red_sloped_concrete_floor_with_central_drain_and_stationary_scraper);
}
when ’sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading’ {
1 - Tech(red_sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading);
}
when ’sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading_plus_shower’
1 - Tech(red_sloped_concrete_floor_with_central_drain_and_stationary_scraper_plus_windbreak_and_shading_plus_
}
default {
1;
}
}
```

share_excretion Share of excretion on the yard according the stay on yard. If the basic feeding is on the yard the share_basicfeeding is assumed. Otherwise the share is selected according to the access.

```
given ( In(exercise_yard) ) {
when 'available_roughage_is_not_supplied_in_the_exercise_yard' {
Tech(share_available_roughage_is_not_supplied_in_the_exercise_yard);
}
when 'available_roughage_is_partly_supplied_in_the_exercise_yard' {
Tech(share_available_roughage_is_partly_supplied_in_the_exercise_yard);
}
when 'available_roughage_is_exclusively_supplied_in_the_exercise_yard' {
Tech(share_available_roughage_is_exclusively_supplied_in_the_exercise_yard);
}
when 'not_available' {
0;
}
}
```

share_excretion_with_grazing Share of excretion on the yard according the stay on yard with parallel acess to Pasture. If the basic feeding is on the yard the share_basicfeeding is assumed. Otherwise the share is selected according to the access.

```
if( (Out(days_with_grazing_and_yard) > 0) and (Out(share_excretion) > Tech(share_available_roughage_is_partly
writeLog({
en => "The category \"roughage is exclusively supplied in the exercise yard\" is not compatible with the numb
de => "Verabreichung von Grundfutter ausschliesslich auf dem Laufhof ist nicht möglich mit den eingegebenen W
fr => "La distribution de fourrage exclusivement dans le parcours extérieur "
. "pour les vâches laitières n'est pas possible pendant les jours de pâturage !\n",
});
return Tech(share_available_roughage_is_partly_supplied_in_the_exercise_yard);
} else {
return Out(share_excretion);
}
```

n_into_yard Annual N excretion on yard for a defined animal category.

```
Val(n_excretion, Excretion) *
(
Out(share_excretion) *
(In(yard_days) - Out(days_with_grazing_and_yard)) / 365 +
Out(share_excretion_with_grazing) *
Out(days_with_grazing_and_yard) / 365
);
```

tan_into_yard Annual soluble N excretion on yard for a defined animal category.

```
Val(tan_excretion,Excretion) *
(
Out(share_excretion) *
(In(yard_days) - Out(days_with_grazing_and_yard)) / 365 +
Out(share_excretion_with_grazing) *
Out(days_with_grazing_and_yard) / 365
);
```

ef_nh3_nyard NH3 emission factor for dairy cow yard.

```
Tech(er_yard) *
Out(c_floor_properties_exercise_yard_SHL) *
Out(c_floor_properties_exercise_yard_LU) *
Out(c_free_factor_yard);
```

nh3_nyard Annual NH3 emission from yard.

```
Out(tan_into_yard) *
Out(ef_nh3_nyard);
```

n_outyard_liquid Annual N flux from liquid part out of yard.

```
Out(n_into_yard) - Out(nh3_nyard);
```

tan_outyard_liquid Annual N flux as TAN from liquid part out of yard into storage.

```
Out(tan_into_yard) - Out(nh3_nyard);
```

n_outyard_solid Annual N flux from solid part out of yard.

```
0;
```

tan_outyard_solid Annual N flux as TAN from solid part out of yard into storage.

```
0;
```

1.105 Livestock::DairyCow::GrazingInput

Input parameters for Grazing, as it is used also in Yard in separete Module.

1.105.1 Inputs

grazing_days Average grazing days per year.

grazing_hours Average grazing hours per day.

1.105.2 Outputs

grazing_hours Grazing hours per day.

```
In(grazing_hours);
```

grazing_days Grazing days per year.

```
In(grazing_days);
```

1.106 Livestock::DairyCow::Housing::CFreeFactor

TODO

1.106.1 Inputs

free_correction_factor Factor to define free.

1.106.2 Outputs

c_free_factor_housing Free correction factor for NH3 housing emission.

```
if ( In(free_correction_factor) // 0 != 0 ) {
writeLog({
en => "You have entered an additional emission mitigation measure for a dairy cows housing of"
. In(free_correction_factor) . "%\!\n" ,
de => "Sie haben eine zusätzliche emissionsmindernde Massnahme für einen Stall der Milchkühe von "
. In(free_correction_factor) . "\% eingegeben!\n" ,
fr => "Vous avez introduit une mesure supplémentaire limitant les émissions provenant d'une "
. "stabulation pour vâches laitières de " . In(free_correction_factor) . "\% !\n"
});
return 1 - In(free_correction_factor) / 100;
}
else {
return 1;
}
```

1.107 Livestock::DairyCow::Housing::Floor

This submodul calculates the annual NH3 reduction due to a grooved floor in housing systems according to the UNECE guideline 2007.

1.107.1 References

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. ECE/EB.AIR/WG.5/2007/13.

1.107.2 Inputs

UNECE_category_1_mitigation_options_for_housing_systems_for_dairy_cows UNECE category 1 mitigation options for housing systems for other cattle (none, toothed scrapper running over a grooved floor).

mitigation_options_for_housing_systems_for_dairy_cows_floor Mitigation options floor for loose housing systems for dairy cows.

1.107.3 Outputs

c_UNECE Correction factor for the emission due to the use of a grooved floor in housing systems.

```
return 1 unless defined In(UNECE_category_1_mitigation_options_for_housing_systems_for_dairy_cows);
given ( In(UNECE_category_1_mitigation_options_for_housing_systems_for_dairy_cows) ) {
when 'raised_feeding_stands' {
1 - Tech(red_raised_feeding_stands);
}
when 'floor_with_cross_slope_and_collection_gutter' {
1 - Tech(red_floor_with_cross_slope_and_collection_gutter);
}
when 'floor_with_cross_slope_and_collection_gutter_and_raised_feeding_stands' {
1 - Tech(red_floor_with_cross_slope_and_collection_gutter_and_raised_feeding_stands);
}
when 'toothed_scrapper_running_over_a_grooved_floor' {
1 - Tech(red_toothed_scrapper_running_over_a_grooved_floor);
}
default {
# add when "none" & default Warning!
1;
}
}
```

c_housing_floor Correction factor for the emission due to the use of housing system adaptations.

```
return 1 unless defined In(mitigation_options_for_housing_systems_for_dairy_cows_floor);
given ( In(mitigation_options_for_housing_systems_for_dairy_cows_floor) ) {
when 'toothed_scrapper_running_over_a_grooved_floor' {
1 - Tech(red_toothed_scrapper_running_over_a_grooved_floor);
}
when 'sloped_concrete_floor_with_drain_and_stationary_scarp_cleaning_every_two_hours' {
1 - Tech(red_sloped_concrete_floor_with_drain_and_stationary_scarp_cleaning_every_two_hours);
}
when 'system_Comfort_Slat_Mat_with_Mat_and_Valve' {
1 - Tech(red_system_Comfort_Slat_Mat_with_Mat_and_Valve);
}
default {
```

```
1;
}
}
```

1.108 Livestock::DairyCow::Housing::KArea

It is assumed that additional areas in stable, as example if the stabel is built for more animal, are fully used. So the emissions increase 5
the double of the regularized minimal area.

TODO (All): Provide Articles which confirm/justify the assumption. As decided on workshop from 2.11.2007.

1.108.1 Inputs

dimensioning_barn Number of available animal places.

1.108.2 Outputs

dimensioning_barn Number of anymal places (needed in other modules).

```
In(dimensioning_barn);
```

area_increase Factor on what barn size does increase the regularized minimal, limited to 0.5

```
if ( (Val(animals, ..::Excretion) < In(dimensioning_barn) ) and (Val(animals, ..::Excretion) != 0) ) {
if ( In(dimensioning_barn) >= (Val(animals, ..::Excretion) * 1.5) ) {
0.5;
}
else {
In(dimensioning_barn) / Val(animals, ..::Excretion) - 1;
}
}
else {
0.0;
}
```

c_area Correction factor for area per animal.

```
1 + (Out(area_increase) * Val(k_area, Type));
```

1.109 Livestock::DairyCow::Housing::Type

This process selects the correction factor for the specific housing types for dairy cows. Among these correction factors are the emission rate for the housing type, the correction factor for the housing type area, the correction factor for the area per animal, the liquid and solid share for the housing type as well as the factor on what barn size increases the regularized minimal (limited to 0.5).

1.109.1 Inputs

housing_type Type of housing.

feeding_boxes Are feeding boxes available?

1.109.2 Outputs

housing_type Housing type (needed in other modules).

```
if ( not defined In(housing_type) ) {
writeLog({
en => "TODO: dairy cow input 'housing_type' not defined!",
de => "TODO: dairy cow input 'housing_type' not defined!",
fr => "TODO: dairy cow input 'housing_type' not defined!"
});
}
In(housing_type);
```

k_area Correction factor for the housing type area.

```
given ( In(housing_type) ) {
return Val(k_area,Type::Tied_Housing_Slurry)                    when 'Tied_Housing_Slurry';
return Val(k_area,Type::Tied_Housing_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manure';
return Val(k_area,Type::Loose_Housing_Slurry)                   when 'Loose_Housing_Slurry';
return Val(k_area,Type::Loose_Housing_Slurry_Plus_Solid_Manure) when 'Loose_Housing_Slurry_Plus_Solid_Manure'
return Val(k_area,Type::Loose_Housing_Deep_Litter)              when 'Loose_Housing_Deep_Litter';
}
```

er_housing Emission rate for the housing type.

```
given ( In(housing_type) ) {
return Val(er_housing,Type::Tied_Housing_Slurry)                    when 'Tied_Housing_Slurry';
return Val(er_housing,Type::Tied_Housing_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Manu
return Val(er_housing,Type::Loose_Housing_Slurry)                   when 'Loose_Housing_Slurry';
return Val(er_housing,Type::Loose_Housing_Slurry_Plus_Solid_Manure) when 'Loose_Housing_Slurry_Plus_Solid_Mar
return Val(er_housing,Type::Loose_Housing_Deep_Litter)              when 'Loose_Housing_Deep_Litter';
}
```

share_liquid Liquid share for the housing type.

```
given ( In(housing_type) ) {
return Val(share_liquid,Type::Tied_Housing_Slurry)                    when 'Tied_Housing_Slurry';
return Val(share_liquid,Type::Tied_Housing_Slurry_Plus_Solid_Manure)  when 'Tied_Housing_Slurry_Plus_Solid_Ma
return Val(share_liquid,Type::Loose_Housing_Slurry)                   when 'Loose_Housing_Slurry';
return Val(share_liquid,Type::Loose_Housing_Slurry_Plus_Solid_Manure) when 'Loose_Housing_Slurry_Plus_Solid_M
return Val(share_liquid,Type::Loose_Housing_Deep_Litter)              when 'Loose_Housing_Deep_Litter';
}
```

c_feeding_boxes Correction factor for feeding boxes

```
if ( defined In(feeding_boxes) and In(feeding_boxes) eq 'yes' ) {
if ( (In(housing_type) eq 'Tied_Housing_Slurry') or (In(housing_type) eq 'Tied_Housing_Slurry_Plus_Solid_Manu
writeLog({
en=>"Feeding boxes are applicable for loose housing systems only!",
de=>"Fütterungsbuchten sind nur für Laufställe anwendbar!",
fr=>"Franz.: Fütterungsbuchten sind nur für Laufställe anwendbar!"
});
return 1;
}
else{
1 - Tech(red_feeding_boxes);
}
}
else {
1;
}
```

1.110 Livestock::DairyCow::Housing::Type::Loose_Housing_Deep_Litter

This process describes the correction factors for the loose housing deep litter system for dairy cows such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.110.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. UNECE executive body for the convention on long-range transboundary air pollution, Working Group on Strategies and Review, EB.AIR/WH.5/2007/13/ 16 July 2007, Genf. 35p.

1.110.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.111 Livestock::DairyCow::Housing::Type::Loose_Housing_Slurry_Plus_Solid_Manure

This process describes the correction factors for the loose housing liquid solid system for dairy cows such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.111.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. UNECE executive body for the convention on long-range transboundary air pollution, Working Group on Strategies and Review, EB.AIR/WH.5/2007/13/ 16 July 2007, Genf. 35p.

1.111.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.112 Livestock::DairyCow::Housing::Type::Loose_Housing_Slurry

This process describes the correction factors for the loose housing slurry system for dairy cows such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.112.1 References:

UNECE 2007. Guidance document on control techniques for preventing and abating emissions of ammonia. UNECE executive body for the convention on long-range transboundary air pollution, Working Group on Strategies and Review, EB.AIR/WH.5/2007/13/ 16 July 2007, Genf. 35p.

1.112.2 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.113 Livestock::DairyCow::Housing::Type::Tied_Housing_Slurry_Plus_Solid_Manure

This process describes the correction factors for the tied housing liquid solid system for dairy cows such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.113.1 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.114 Livestock::DairyCow::Housing::Type::Tied_Housing_Slurry

This process describes the correction factors for the tied housing slurry system for dairy cows such as the housing specific emission rate, the area per animal correction, the liquid share and solid share.

1.114.1 Outputs

er_housing Emission rate for specific housing type.

```
Tech(er);
```

share_liquid Part of Ntot flowing into liquid storage for selected housing type.

```
Tech(share_liquid);
```

k_area Correction factor for area per animal.

```
Tech(k_area);
```

1.115 Livestock::DairyCow::Excretion

This process calculates the annual N excretion (total N and Nsol (urea plus measured total ammoniacal nitrogen)) of a number of dairy cows as a function of the milk yield and the supplied feed ration. Nitrogen surpluses from increased nitrogen uptake are primarily excreted as Nsol in the urine. Eighty percent of the increased N excretion is therefore added to the Nsol fraction.

The standard N excretion was taken from the official Swiss fertilizer guidelines. These values were compiled on the basis of official feeding recommendations (RAP 1999) by a group of feeding experts under the lead of H. Menzi. Even though the methodology used is not documented in detail, it was well known to the authors of DYNAMO.

1.115.1 References:

Burgos SA, Robinson PH, Fadel JG, DePeters EJ 2005. Ammonia volatilization potential: Prediction of urinary urea nitrogen output on lactating dairy cows. Agriculture, Ecosystems and Environment 111:261-269.

Peterson SO, Sommer SG, Aaes O, Soegaard K 1998. Ammonia losses from urine and dung of grazing cattle: effect of N intake. Atmospheric environment 32:295-300.

RAP 1999. Fütterungsempfehlungen und Nährwerttabelle für Wiederkäuer. 4. Überarbeitete Auflage, 327p, Landwirtschaftliche Lehrmittelzentrale, Zollikofen.

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.115.2 Inputs

animalcategory Animal category

dairy_cows Number of dairy cows in barn.

inp_n_excretion Annual standard N excretion for a dairy cow

1.115.3 Outputs

animals Number of dairy cows in barn.

```
In(dairy_cows);
```

animalcategory Animal category

```
In(animalcategory);
```

n_excretion_animal Annual mean total N excreted per animal.

```
if ( lc In(inp_n_excretion) eq 'standard' ) {
Tech(standard_N_excretion) *
Val(cmilk_yield, Excretion::CMilk) *
Val(c_feed_ration, Excretion::CFeed);
} else {
if ( (In(inp_n_excretion) < 0.25 * Tech(standard_N_excretion)) or
(In(inp_n_excretion) > 1.75 * Tech(standard_N_excretion)) ) {
writeLog({
en => "TODO: Die eingegebene N-Ausscheidung für Milchkühe weicht im mehr als 75% vom Standard ab.",
de => "Die eingegebene N-Ausscheidung für Milchkühe weicht im mehr als 75% vom Standard ab.",
fr => "TODO: Die eingegebene N-Ausscheidung für Milchkühe weicht im mehr als 75% vom Standard ab.",
});
}
```

```
In(inp_n_excretion);
};
```

n_excretion Annual total N excreted by the specified number of animals.

```
Out(n_excretion_animal) *
Out(animals);
```

tan_excretion Annual soluble N excreted by the specified number of animals.

```
my $exc = Tech(standard_N_excretion);
$exc = Out(n_excretion_animal) /
Val(cmilk_yield, Excretion::CMilk) /
Val(c_feed_ration, Excretion::CFeed) unless lc In(inp_n_excretion) eq 'standard';
# n related to milk yield
my $exc_milkyield = $exc * Out(animals) * Val(cmilk_yield, Excretion::CMilk);
# tan related to milk yield
$exc_milkyield * Tech(share_Nsol) +
# tan related to feeding
(Out(n_excretion) - $exc_milkyield) * Tech(feed_influence_on_Nsol);
```

1.116 Livestock::DairyCow::Excretion::CFeed

This process accounts for the fact, that special rations can result in higher or lower N excretions as compared to standard excretions from Flisch et al. (2009). A differentiated consideration of the duration of the summer and winter feeding period according to farm location (altitude etc.) is possible but was not implemented for the emission inventory.

1.116.1 References:

Flisch R, Sinaj S, Charles R, Richner W 2009. Grundlagen für die die Düngung im Acker- und Futterbau. Agrarforschung 16(2).

1.116.2 Outputs

c_feed_ration Feed ration correction factor for annual N excretion.

```
1 +
Val(c_summer_ration, CFeedSummerRatio) +
Val(c_winter_ration, CFeedWinterRatio) +
Val(c_concentrates_summer, CConcentrates) +
Val(c_concentrates_winter, CConcentrates);
```

1.117 Livestock::DairyCow::Excretion::CConcentrates

This formula takes into account the amount of concentrates used per cow during the winter and summer feeding period. The correction is based on the fact that concentrates (grains) can specifically balance the energy to protein ratio, thus reducing the crude protein.

1.117.1 References:

Flückiger E 1989. Stickstoff- und Mineralstoffumsatz von Milchkühen in Abhängigkeit von Rationentyp und Produktionsphase unter besonderer Berücksichtigung umweltrelevanter Aspekte. Diss ETH Nr 8865.

TODO (Harald Menzi): Confirm calculation, and standard ratios.

1.117.2 Inputs

amount_summer Amount of concentrates per animal per day in summer.

amount_winter Amount of concentrates per animal and per day in winter.

1.117.3 Outputs

amount_summer Amount of concentrates per animal and per day in summer

```
In(amount_summer);
```

amount_winter Amount of concentrates per animal and per day in winter

```
In(amount_winter);
```

c_concentrates_summer Calculation of correction to excretions with 0.3, 0.5, 1, 2, 3, 4, 6 kg of concentrate (barley and wheat) per cow per day, taking into account the amount of roghage substituted by the concentrate (according to the energy content) and the relative differnce of the protein content of the roughage and the concentrate. Results used to calculate a regression for % modification of N excretion due to concentrate quantity summer. Standard for Summer = 1 kg animal-1 day-1.

```
Tech(par_a_summer) + Tech(par_b_summer) * Out(amount_summer);
```

c_concentrates_winter Calculation of correction to excretions with 0.3, 0.5, 1, 2, 3, 4, 6 kg of concentrate (barley and wheat) per cow per day, taking into account the amount of roghage substituted by the concentrate (according to the energy content) and the relative differnce of the protein content of the roughage and the concentrate. Results used to calculate a regression for % modification of N excretion due to concentrate quantity summer. Standard for Winter = 2 kg animal-1 day-1.

```
Tech(par_a_winter) +  Tech(par_b_winter) * Out(amount_winter);
```

1.118 Livestock::DairyCow::Excretion::CFeedWinterRatio

This process calculates the correction factor for the N excretion during the winter feeding period as compared to the standard excretion values of Walther et al. (2001). The average feed ration considered for the standard excretion presented in Walther et al. (2001) was calculated as average of four summer and four winter standard rations using dsummer and dwinter mentioned above. To calculate the N excretion of farm-specific winter rations, excretions were calculated with the same model used by the authors of Walther et al. (2001) (based on official feeding recommendations (RAP 1999)) using proportions of the specific feed typically used on farms (expert assumptions). No correction was considered for hay, because hay is used by virtually all farms during the winter feeding period. The thus calculated winter excretions were then expressed as % of average excretions.

1.118.1 References:

RAP 1999. Fütterungsempfehlungen und Nährwerttabellen für Wiederkäuer. 4. überarbeitete Auflage. LMZ Zollikofen. Walther U,Ryser JP, Flisch R (Eds.) 2001. Grundlagen für die Düngung im Acker- und Futterbau 2001. Agrarforschung 8:1-80.

1.118.2 Inputs

share_maize_silage_winter Proportion of animals receiving maize silage in winter.

share_grass_silage_winter Proportion of animals receiving grass silage in winter.

share_maize_pellets_winter Proportion of animals receiving maize pellets in winter.

share_potatoes_winter Proportion of animals receiving potatoes in winter.

share_beets_winter Proportion of animals receiving beets in winter.

1.118.3 Outputs

share_grass_silage_winter Share

```
if ( In(share_grass_silage_winter) > 1 ) {
In(share_grass_silage_winter) / 100;
} else {
In(share_grass_silage_winter);
}
```

share_maize_silage_winter Share

```
if ( In(share_maize_silage_winter) > 1 ) {
In(share_maize_silage_winter) / 100;
} else {
In(share_maize_silage_winter);
}
```

share_maize_pellets_winter Share

```
if ( In(share_maize_pellets_winter) > 1 ) {
In(share_maize_pellets_winter) / 100;
} else {
In(share_maize_pellets_winter);
}
```

share_potatoes_winter Share

```
if ( In(share_potatoes_winter) > 1 ) {
In(share_potatoes_winter) / 100;
} else {
In(share_potatoes_winter);
}
```

share_beets_winter Share

```
if ( In(share_beets_winter) > 1 ) {
In(share_beets_winter) / 100;
} else {
In(share_beets_winter);
}
```

share_hay_only_winter Share

```
if ( Out(share_grass_silage_winter) >= Out(share_maize_silage_winter) and
Out(share_grass_silage_winter) >= Out(share_maize_pellets_winter) ) {
1 - Out(share_grass_silage_winter);
} elsif ( Out(share_maize_silage_winter) >= Out(share_grass_silage_winter) and
Out(share_maize_silage_winter) >= Out(share_maize_pellets_winter) ) {
1 - Out(share_maize_silage_winter);
} else {
1 - Out(share_maize_pellets_winter);
}
```

c_winter_ration Winter feed ration correction factor for annual N excretion.

```
Tech(c_grass_silage_winter) * Out(share_grass_silage_winter) +
Tech(c_maize_silage_winter) * Out(share_maize_silage_winter) +
Tech(c_maize_pellets_winter) * Out(share_maize_pellets_winter) +
Tech(c_potatoes_winter) * Out(share_potatoes_winter) +
Tech(c_beets_winter) * Out(share_beets_winter) +
Tech(c_default_hay) * Out(share_hay_only_winter);
```

1.119 Livestock::DairyCow::Excretion::CFeedSummerRatio

This process calculates the correction factor for N excretion during the summer feeding period as compared to the standard excretion values of Walther et al. (2001). The average feed ration considered for the standard excretion presented in Walther et al. (2001) was calculated as average of four summer and four winter standard rations using dsummer and dwinter mentioned above. To calculate the N excretion of farm-specific summer rations, excretions were calculated with the same model used by the authors of Walther et al. (2001) (based on official feeing recommendations (RAP 1999)) using proportions of the specific feed typically used on farms (expert assumptions). No correction was considered for grass and grass silage, because grass is used by virtually all farms during the summer feeding period und because the crude protein content of grass silage is not much lower than that of grass. The thus calculated summer excretions were then expressed as % of average excretions.

1.119.1 References:

RAP 1999. Fütterungsempfehlungen und Nährwerttabellen für Wiederkäuer. 4. überarbeitete Auflage. LMZ Zollikofen. Walther U, Ryser JP, Flisch R (Eds.) 2001. Grundlagen für die Düngung im Acker- und Futterbau 2001. Agrarforschung 8:1-80.

1.119.2 Inputs

share_hay_summer Proportion of animals receiving hay in summer.

share_maize_silage_summer Proportion of animals receiving maize silage in summer.

share_maize_pellets_summer Proportion of animals receiving maize pellets in summer.

1.119.3 Outputs

share_hay_summer Share

```
if (In(share_hay_summer) > 1) {
In(share_hay_summer) / 100;
} else {
In(share_hay_summer);
}
```

share_maize_silage_summer Share

```
if ( In(share_maize_silage_summer) > 1 ) {
In(share_maize_silage_summer) / 100;
} else {
In(share_maize_silage_summer);
}
```

share_maize_pellets_summer Share

```
if (In(share_maize_pellets_summer) > 1) {
In(share_maize_pellets_summer) / 100;
} else {
In(share_maize_pellets_summer);
}
```

share_grass_only_summer Share

```
if ( Out(share_hay_summer) >= Out(share_maize_silage_summer) and
Out(share_hay_summer) >= Out(share_maize_pellets_summer) ) {
1 - Out(share_hay_summer);
} elsif ( Out(share_maize_silage_summer) >= Out(share_hay_summer) and
Out(share_maize_silage_summer) >= Out(share_maize_pellets_summer) ) {
1 - Out(share_maize_silage_summer);
} else {
1 - Out(share_maize_pellets_summer);
}
```

c_summer_ration Summer feed ration correction factor for annual N excretion.

```
Tech(c_hay_summer) * Out(share_hay_summer) +
Tech(c_maize_silage_summer) * Out(share_maize_silage_summer) +
Tech(c_maize_pellets_summer)  * Out(share_maize_pellets_summer) +
Tech(c_default_grass) * Out(share_grass_only_summer) ;
```

1.120 Livestock::DairyCow::Excretion::CMilk

This process describes the relationship between the milk yield and the N excretion. While the N excretion decreases by 10% per 1000 kg less milk yield below the standard milk yield, nitrogen excretion increases by 5% per 1000 kg more milk yield above the standard milk yield. This correction factor was taken from GRUDAF 2009. It was originally derived from excretion calculations for different milk yields ranging from 4000 to 10000 kg year-1. The lower increase of the nitrogen excretion above 6500 kg results from the increasing proportion of concentrate necessary to cover the energy requirement for yields above 6500 kg. It is thus increasingly possible to reduce the unbalance of energy and protein existing in virtually all rations with a high proportion of roughage.

1.120.1 References:

Flisch R, Sinaj S, Charles R, Richner W (Eds.) 2009. Grundlagen für die die Düngung im Acker- und Futterbau 2009 (GRUDAF), Agrarforschung 16.2.

1.120.2 Inputs

milk_yield Annual milk yield per dairy cow.

1.120.3 Outputs

milk_yield Milk yield.

```
In(milk_yield);
```

cmilk_yield Milk yield correction factor for annual N excretion.

```
if ( Out(milk_yield) > Tech(standard_milk_yield) ) {
1 + (Out(milk_yield) - Tech(standard_milk_yield)) / 1000 * Tech(a_high);
}
else {
```

```
1 + (Out(milk_yield) - Tech(standard_milk_yield)) / 1000 * Tech(a_low);
}
```