CSIRO AGRICULTURE
www.csiro.au

AusFarm User Notes #1
Integrating APSIM Soil and Crop Models into AusFarm

October 2015

www.csiro.au

Enquiries should be addressed to:
Dr Andrew Moore

CSIRO Plant Industry
GPO Box 1600
Canberra ACT 2601
grazplan@csiro.au

Copyright and Disclaimer

Important Disclaimer

Contents

1. INTRODUCTION

AusFarm is a software tool that allows problems to be analysed with simulation models of physical and biological systems. AusFarm is highly generic, but it has been built primarily to assist decision-making in agricultural enterprises at scales ranging from paddocks to whole landscapes.

This document describes how to integrate APSIM soil (SOILWAT and SOILN), surface litter (SURFACEOM) and plant (PLANT and SORGHUM) components into AusFarm. This allows mixed crop-pasture simulation studies to be carried out. It should be read in conjunction with the AusFarm Help file and the document "AusFarm – a Tutorial" that is distributed with the AusFarm software.

This version of the document applies to components released as part of APSIM version 7.5.

2. REQUIRED FILES

Note: If you have set up AusFarm on your computer by using the AusFarm Installer with APSIM support (*setupaf_apsim.exe*), the APSIM files described below will have been installed and configured for you automatically, and you may proceed to Section 4. If you have used the basic AusFarm Installer (*setupaf.exe*) you will need to manually install the APSIM components as described in this and the following section.

- To install one of these APSIM components, the following files are required. Note that all the utility files must be present if any component is to be installed:

Component	Executable File	Context File	Utility Files*
SOILWAT	soilwat.dll	soil.xml	apsimshared.dll
SOILN	soiln.dll	soil.xml	componentinterface.dll
SurfaceOM	surfaceom.dll	surfaceom.xml	componentinterface2.dll
Wheat	plant.dll	wheat.xml	fortrancomponentinterface.dll
Barley	plant.dll	barley.xml	fortrancomponentinterface2.dll
Canola	plant.dll	canola.xml	general.dll
Lupin	plant.dll	lupin.xml	iconv.dll
Chickpea	plant.dll	chickpea.xml	libxml2.dll
Fababean	plant.dll	fababean.xml	
Field Pea	plant.dll	fieldpea.xml	
Sorghum	sorghum.dll	sorghum.xml	

*All the utility files must be present in order to install any of these APSIM components. Versions of these files that are compatible with AusFarm should be obtained directly from the APSIM Initiative.

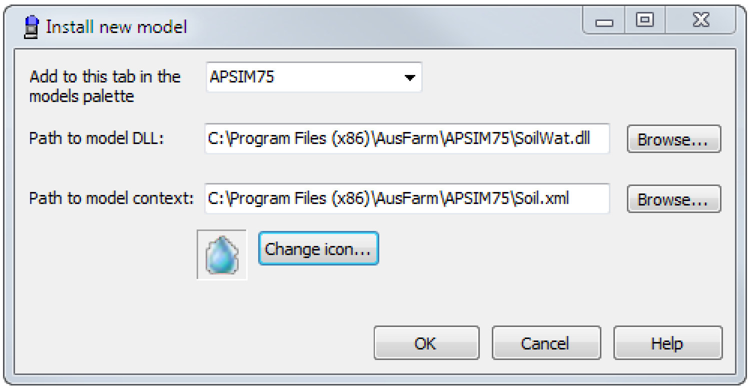
- It is also desirable to have an icon for each component, so that it can be readily identified in the AusFarm palette. Icons should be 16x16 pixels. A set of icons is available upon request from the author.
- The APTRANS component (aptrans.dll) is required. This file, together with a parameter file describing the quality of forage, can be obtained from the author.
- When integrating the PLANT and STOCK components, it is common to use the PASTURE component as a surface residue submodel (this allows the residues to be grazed by stock). A parameter file that permits the PASTURE module to be used in this way can be obtained from the author.

3. INSTALLING NEW COMPONENTS

To install an APSIM component, the following steps must be carried out:

1. Place all the necessary files in a single folder. An "APSIM75" subfolder under the main AusFarm program directory is recommended, e.g. c:\Program Files\AusFarm\APSIM75.

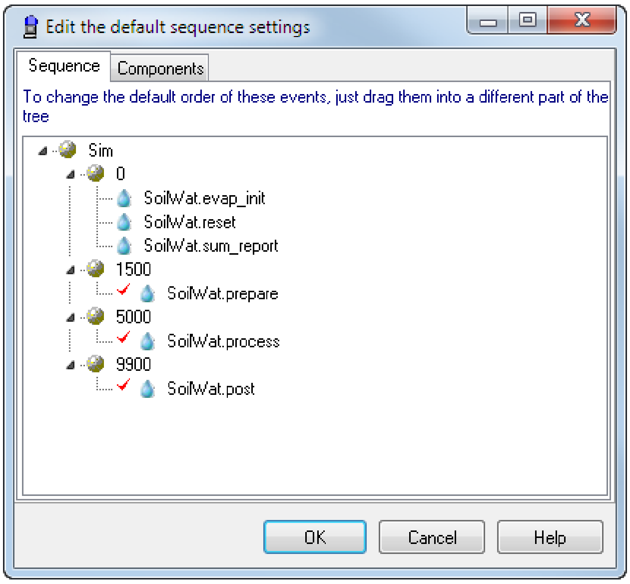
2. Open the AusFarm program and choose the **Model|Install model** menu option. The Install new Model dialog will appear:



   Use the Browse buttons to navigate to the executable file (the "model DLL") and to the context file for the desired component. **Both files must be specified.**

3. If an icon file for this component is available, click the **Change icon…** button. The Change Icon dialog will appear. Click the **Browse…** button on this dialog, navigate to the icon file and select it (you will need to change the file type in the combo box at the bottom of the dialog).

4. Click OK. The next step is to set up the default sequencing for the events of the new component. This is done with the Event Configuration dialog, which will appear automatically:



Component	Event	Sequence Value
SOILWAT	Prepare	1500
	Process	5000
	Post	9900
SURFACEOM	Process	5100
	Post	9910
SOILN	Process	5200
PLANT/SORGHUM	Prepare	1500
	Process	6000

Review the list of events that is shown on the **Sequence** tab of the dialog. The events in the table above should be selected for sequencing (i.e. have non-zero sequence values and red ticks by the event name). If they are not correctly entered, switch to the **Components** tab of the dialog and change them. Be sure to set the check boxes alongside the sequenced events as well as entering their sequence values.

- An icon denoting the newly-installed component will appear on the "Standard" tab of the component palette. You can relocate the icon to another tab of the palette by right-clicking on the icon and selecting the **Move** option on the popup menu that appears.

4. THE APTRANS COMPONENT

Simulations using these APSIM components can be run in AusFarm without further ado. If, however, it is intended to run simulations that combine these components with the GRAZPLAN pasture and ruminant models, then a number of potential difficulties arise:

- The property and event interfaces of the APSIM and GRAZPLAN components are not completely compatible. The parameters of some events in APSIM components are set out in a format (known as "APSIM variant") that requires translation.
- The APSIM PLANT component does not include an interface or logic for the effects of grazing livestock, and the SOILN component does not include an interface for the return of animal excreta.
- Some management activities (notably the addition of N fertilizer) are not represented by events in these components – in APSIM they are implemented as events of "manager" components, which in turn reset the state values of the soil components.

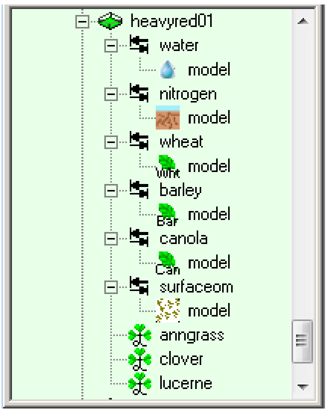
These potential difficulties are resolved by using the APSIM components in conjunction with the APTRANS component. As its name suggests, APTRANS modules translate data from APSIM-variant format into records that can be read by other components. When used in

conjunction with the PLANT component, APTRANS implements logic that makes the plant grazeable by livestock; and when used with the SOILN components, it permits simple N fertilization to take place.

The APTRANS component must be installed in AusFarm before it can be used. The installation process is as described above, except that no context file need be specified and an icon will be provided automatically.

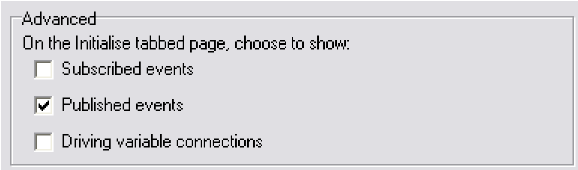
5. ADDING APSIM MODULES TO SIMULATIONS

When adding an APSIM soil or plant component alongside the GRAZPLAN models, the following configuration of the modules should be followed:



- All plant and soil components in a given area of land should be contained within a Paddock module. (The Paddock module implements resource “arbitration” logic.)
- Each APSIM plant or soil component should be placed within an APTRANS module that will manage its interface. The exception is the Sorghum component (see below), which should be placed within a Generic module.
- By convention, the APTRANS (or Generic) module is given a name corresponding to the model logic, while the APSIM module is named `model`. This convention produces a natural naming scheme for both the APTRANS module and the APSIM module (e.g. `paddock3.nitrogen` and `paddock3.nitrogen.model` respectively) when writing management scripts or selecting output variables.

Before proceeding to the initialization of the APSIM plant and soil modules, open the Options dialog and ensure that the published-events list is visible in the Initialise tab of simulation windows:



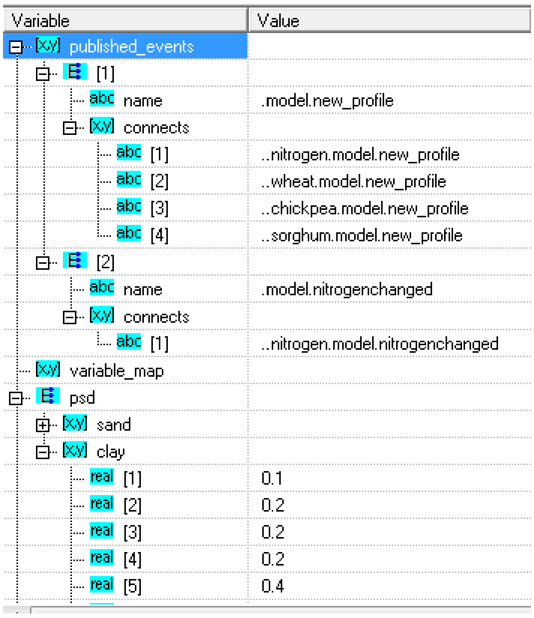
5.1 Paddock modules

The Paddock modules in the simulation need to be initialised as follows:

- If PLANT modules are being used, the `allocator` initialization value for each Paddock module should be set to **3**. This activates a multi-species version of the Monteith (“kl”) water uptake logic.
- As a result, any PASTURE modules in the paddock must also be initialised with values for the `kl` and `ll` properties. The `kl` and `ll` arrays in a PASTURE module must have one element for each layer of the soil water model, except that `ll` may be specified with zero length (in which case the soil lower limit values are used).

5.2 SoilWat modules

To use a SOILWAT module alongside GRAZPLAN modules, its parent **APTRANS** module should be configured as follows:



- The `psd` (particle size distribution) initialization property needs to be populated. This is a record with two array fields, into which the sand and clay content (in g/g) of each soil layer should be entered.
- The SOILWAT module will generate `new_profile` and `NitrogenChanged` events that must be explicitly routed to only those SOILN and PLANT modules that are located in the same paddock. To specify this routing, add an element to the `published_events` initialization array. This array element is a record that should be populated as follows:
  - ⇒ its `name` field is a string value that should be set to the name of the event to be routed. In this case we are routing an event from the SOILWAT module, so the value should be (e.g.) "`.model.new_profile`" assuming that the naming convention described above has been followed.
  - ⇒ its `connects` field is an array of string values. In the case of the `new_profile` event, this array should be sized to have one member for each SOILN or PLANT module in the paddock, and the names of all the receiving event handlers should be entered, e.g. "`..wheat.model.new_profile`". The `NitrogenChanged` event needs to connect only with the SOILN module in its paddock, e.g. "`..nitrogen.model.nitrogenchanged`".

Note: In the `published_events` property, the notation "." at the start of an event name denotes the current module, and ".." denotes its parent module. This notation simplifies matters when paddocks or other modules are being copied and pasted.

The SOILWAT module itself requires no special configuration; its initialization data must be entered in XML format (as in an APSIM "sim" file), within an `<initdata>` XML element, as shown in this partial example:

```
<initdata>

    <!-- APSoil #220: grey Vertosol at Billa Billa; depth extended to 1800 mm  -->
    <!-- Air-dry water contents below 600 mm changed to sensible values  -->

    <!-- Cumul: 150    300    600    900    1200   1500   1800   2100   2400 mm-->
    <dlayer>    150    150    300    300    300    300    300    300    300  </dlayer>
    <bd>      1.480  1.485  1.485  1.540  1.570  1.570  1.570  1.570  1.570  </bd>
    <sat>     0.410  0.410  0.410  0.390  0.380  0.380  0.380  0.380  0.380  </sat>
    <dul>     0.360  0.360  0.360  0.340  0.330  0.330  0.330  0.330  0.330  </dul>
    <ll15>    0.200  0.200  0.200  0.200  0.200  0.200  0.200  0.200  0.200  </ll15>
    <air_dry> 0.100  0.160  0.160  0.160  0.160  0.160  0.160  0.160  0.160  </air_dry>
    <swcon>   0.30   0.30   0.30   0.30   0.30   0.30   0.30   0.30   0.30   </swcon>
    <sw>      0.200  0.200  0.200  0.230  0.240  0.290  0.310  0.330  0.330  </sw>
    <diffus_const> 40.0     ()    </diffus_const>
    <diffus_slope> 16.0     ()    </diffus_slope>
    <cn2_bare>     73       ()    </cn2_bare>
    <cn_red>       20             </cn_red>
    <cn_cov>        0.80          </cn_cov>
    <salb>          0.13    ()    </salb>
    <SummerCona>    3.5           </SummerCona>
    <WinterCona>    2.5           </WinterCona>
    <SummerU>       6.0           </SummerU>
    <WinterU>       4.0           </WinterU>
    <SummerDate>1-nov</SummerDate>
    <WinterDate>1-apr</WinterDate>
```

5.3 SoilN modules

To use a SOILN module in conjunction with GRAZPLAN modules, its parent APTRANS module should be configured as follows:

- `new_solute` events generated by the SOILN module must be explicitly routed to only the SOILWAT module that is located in the same paddock. The `published_events` property should have a single element, populated as follows:
  - ⇒ its `name` field is should be set to the name of the event to be routed, i.e. "`.model.new_solute`" assuming that the naming convention described above has been followed.
  - ⇒ its `connects` field should have a single element. The names of the receiving event should be entered, e.g. "`..water.model.new_solute`".

- When an APSIM SurfaceOM module is present, `actualResidueDecompositionCalculated` events generated by the SOILN module must be explicitly routed to only the SurfaceOM module that is located in the same paddock. The `published_events` property should have a single element, populated as follows:
  - ⇒ its `name` field is should be set to the name of the event to be routed, i.e. "`.model.actualresiduedecompositioncalculated`" assuming that the naming convention described above has been followed.
  - ⇒ its `connects` field should have a single element. The names of the receiving event should be entered, e.g. "`..surfaceom.model.actualresiduedecompositioncalculated`".

- When configured with a child SOILN module, the APTRANS component executes a simple model for the dynamics of excreta returned to the soil by grazing animals. The `excrete_params` array gives the values of three parameters that govern this model:
  - ⇒ Element 1 is the proportion of excreta that is assumed to be deposited in "camp" areas and hence lost to the system.
  - ⇒ Element 2 is the proportion of urine that is assumed to volatilize, with the result that the urea-N it contains is lost to the system.
  - ⇒ Element 3 is a reference breakdown rate for faecal organic matter on the soil surface (d^{-1}).

  The `excrete_params` property may be set at zero length, in which case the default values [0.30,0.25,0.035] are used.

The SOILN module itself requires no special configuration; its initialization data must be entered in XML format (as in an APSIM "sim" file), within an `<initdata>` XML element.

```
<initdata>
   <!-- APSoil #220: grey Vertosol at Billa Billa; depth extended to 1800 mm  -->

    <!-- Cumul: 150    300    600    900    1200   1500   1800   2100   2400 mm-->
    <oc>       1.24   1.24   1.25   1.10   0.71   0.34   0.26   0.15   0.15 </oc>
    <ph>       8.0    8.7    8.9    8.9    8.6    7.6    7.6    7.6    7.6  </ph>
    <fbiom>    0.04   0.02   0.02   0.02   0.01   0.01   0.01   0.01   0.01 </fbiom>
    <finert>   0.40   0.50   0.50   0.60   0.60   0.70   0.80   0.80   0.80 </finert>
    <ureappm>  0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0  </ureappm>
    <no3ppm>  30.0   10.0    4.0    2.0    1.5    1.5    1.5    1.5    1.5  </no3ppm>
    <nh4ppm>   0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0    0.0  </nh4ppm>
    <root_cn>      40     </root_cn>
    <root_wt>    1000     </root_wt>
    <soil_cn>      12.0   </soil_cn>
    <enr_a_coeff>   7.4   </enr_a_coeff>
    <enr_b_coeff>   0.2   </enr_b_coeff>
    <profile_reduction>off</profile_reduction>
```

5.4 SurfaceOM modules

The APTRANS component that is managing a SurfaceOM module must have its "surfaceom_types" array set to the same list of OM types (and in the same order) that is initialised in the SurfaceOM module.

If residues are to be grazed by livestock, then a parameter set file (the same as that used for the grazing of APSIM-Plant modules) must be specified using the "forage_params" initialisation value.

The SurfaceOM module will generate `incorpfompool`, `nitrogenchanged`, and `potentialresiduedecompositioncalculated` events that must be explicitly routed to only the SOILN module that is located in the same paddock. The `published_events` property should have an element for each of these events, populated as follows:

- ⇒ its `name` field is should be set to the name of the event to be routed, e.g. "`.model.incorpfompool`" assuming that the naming convention described above has been followed.
- ⇒ its `connects` field should have a single element. The name of the receiving event should be entered, e.g. "`..nitrogen.model.incorpfompool`".

Variable	Value
published_events	
[1]	
name	.model.potentialresiduedecompositioncalculated
connects	
[1]	..nitrogen.model.potentialresiduedecompositioncalculated
[2]	
name	.model.incorpfompool
connects	
[1]	..nitrogen.model.incorpfompool
[3]	
name	.model.nitrogenchanged
connects	
[1]	..nitrogen.model.nitrogenchanged
variable_map	
psd	
excrete_params	
plant_type	<none>
forage_params	C:\APSIMsrc\Test simulations\Library Crop Quality.prm
kno3	0.0
knh4	0.0
start_nstress	4.3
surfaceom_types	

When configured with a child SURFACEOM module, the APTRANS component implements a model for the digestibility of various plant parts. The `forage_params` property gives the name of a file containing parameters for this submodel.

The SURACEOM module itself requires no special configuration, unless it is being used in conjunction with the GRAZPLAN pasture model. In that case, an OM type named "pasture" must be defined. Recommended parameters are:

```
<pasture>
    <derived_from>base_type</derived_from>
    <fom_type>pasture</fom_type>
    <specific_area description="specific area of residue (ha/kg)">0.0004  </specific_area>
</pasture>
```

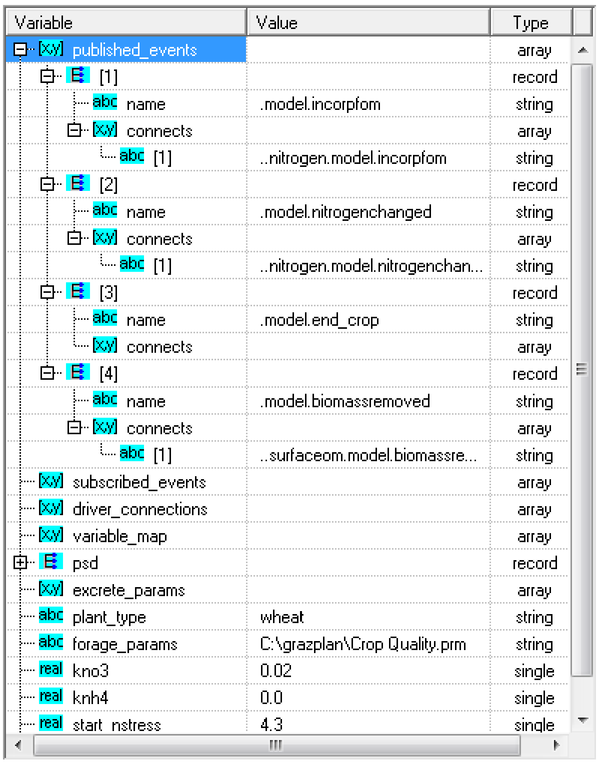
Its initialization data must be entered in XML format (as in an APSIM "sim" file), within an `<initdata>` XML element.

When the GRAZPLAN Pasture component is used in conjunction with APSIM-SurfaceOM, it will transfer litter to that SurfaceOM module at the time it falls and the decomposition and breakdown logic of APSIM-SurfaceOM will be used in the simulation. As a result, the mass of litter reported by the Pasture modules will always be zero in this configuration.

5.5 Plant modules

To use a PLANT module (wheat, canola, lupin etc, but *not* sorghum) alongside GRAZPLAN modules, its parent APTRANS module should be configured as follows:

- The plant_type initialization property must be set to match the crop_type initialization property of the PLANT module.
- The APTRANS module will calculate the N uptake by the plants. The values of `kno3` and `knh4` must therefore be set to appropriate values for the relevant crop.
  - ⇒ If both are set to zero, then a "mass flow+diffusion" model logic will be used. This is the usual configuration for oilseeds and grain legumes in APSIM.
  - ⇒ Otherwise `kno3` and `knh4` give the maximum proportions of available nitrate-N and ammonium-N respectively that can be taken up. This is the usual configuration for cereals in APSIM; default values are `kno3=0.02` and `knh4=0.0`.
- The `start_nstress` value must be set to the phenological stage before which N stress is assumed not to affect growth (4.3 is the usual value in APSIM).
- When configured with a child PLANT module, the APTRANS component implements a model for the digestibility of various plant parts and an interface that permits the crop to be grazed by livestock. The `forage_params` property gives the name of a file containing parameters for this submodel. If the PLANT module is to be used in conjunction with a PASTURE module for surface residues or if it is to be grazed, then this parameter file must be specified.
- There are four events published by the PLANT module that must be explicitly routed using the `published_events` property of the parent APTRANS module:
  - ⇒ `incorpfom` and `nitrogenchanged` events must be routed to the SOILN module in the same paddock;
  - ⇒ `end_crop` events published by the PLANT module must be prevented from reaching any other modules. This is done by setting the `connects` field corresponding to the `end_crop` event to have zero length (as in the figure above);
  - ⇒ `biomassremoved` events must be routed from the PLANT component to a SurfaceOM module.



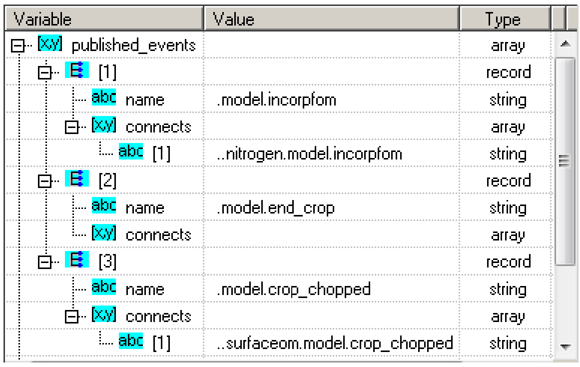
In order for the PLANT module to operate successfully, its initialization data must include the following:

- The `uptake_source` initialization property must be added and set to "`apsim`". This denotes that water and nitrogen uptakes are computed externally to the module:
<uptake_source>apsim</uptake_source>
- The equation that describes the effect of defoliation on phenology must be disabled, since an alternative equation is implemented in the APTRANS component as part of its grazing interface. To do this, the following initialization properties should be set:
<x_removebiompheno>0.0   1.0   ()</x_removebiompheno>
<y_removefractpheno>0.0 0.0   ()</y_removefractpheno>

5.6 Sorghum modules

To use a SORGHUM module alongside GRAZPLAN modules, it should be nested within a Generic module (as described above).

Sorghum modules must be managed as monocultures, as the SORGUM component does not implement the interfaces necessary to allow the arbitration logic from Paddock to be employed. It is also not currently possible to implement grazing of the SORGHUM component (dead sorghum residues can, however, be grazed via the interface to the SurfaceOM component).



- There are three events published by a SORGHUM module that must be explicitly routed using the `published_events` property of the parent Genericmodule:
  - ⇒ `incorpfom` events must be routed to the SOILN module in the same paddock;
  - ⇒ `end_crop` events published must be prevented from reaching any other modules. This is done by setting the `connects` field corresponding to the `end_crop` event to have zero length (as in the figure at right);
  - ⇒ `crop_chopped` events must be routed from the Sorghum module to a Pasture component that is acting as a surface residue submodel.

In order for the SORGHUM module to operate successfully, its initialization data must include the following:

- The `uptake_source` initialization property must be left unspecified. This denotes that water and nitrogen uptakes are computed internally to the module.

6. MANAGEMENT SCRIPTS

The parameters of some events in APSIM components are passed in the complex APSIM-variant format. The APTRANS component is designed to hide this from the user by accepting events containing the same information, translating them into the format used by the APSIM component, and passing them on. The main such event likely to be used in AusFarm

simulations is the `irrigated` event in the SoilWat module. An example of its use is given below; the values for "sender" and "sender_id" should be present, but do not need to be correct.

All other events should be sent to the APSIM component directly.

The APTRANS component also provides a few additional events:

- When used in conjunction with a PLANT component, the APTRANS component has its own `proxy_harvest` event that distinguishes residues that are cut (and hence become litter) from those below cutting height (which become standing residues).
- When used in conjunction with a SOILN component, the APTRANS component has a `fertilize` event that applies inorganic N fertilizer.

Hence (assuming that the naming conventions described above are used):

```
paddock2.wheat.sow      crop_class='plant', cultivar='Janz', plants=200.0, ...
                        sowing_depth=50.0, row_spacing=150.0

paddock2.wheat.proxy_harvest cut_height=300.0


paddock2.wheat.model.kill_crop
paddock2.wheat.model.end_crop

! Irrigate with 25 mm. The correct value for sender_id can be found from the
! message log
paddock2.water.irrigated sender='manager', sender_id=4, amount=25.0

paddock2.water.model.reset

! Fertilize with ammonium nitrate
paddock2.soiln.fertilize  no3_n=25.0, nh4_n=25.0, urea_n=0.0


! Define a data structure for use with the incorpfom event
define layer_val = [ (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0),
                     (FOM:(amount:0.0; C:0.0; N:0.0; P:0.0; AshAlk:0.0); CNR:0.0; LabileP:0.0)]

! Put some data values into the structure (only one is shown here)
set layer_val[1]:FOM:amount = 4.8

paddock2.soiln.model.incorpfom Type='wheat', Layer=layer_val

paddock2.soiln.model.reset
```

Contact Us
Phone: 1300 363 400
+61 3 9545 2176
Email: enquiries@csiro.au
Web: www.csiro.au/flagships

CSIRO and the Flagships program

Australia is founding its future on science and innovation. Its national science agency, CSIRO, is a powerhouse of ideas, technologies and skills. CSIRO initiated the National Research Flagships to address Australia's major research challenges and opportunities. They apply large scale, long term, multidisciplinary science and aim for widespread adoption of solutions.