
traDSSAT Documentation

Version 0.1.4

Julien Malard,

déc. 31, 2020

Table des matières

| | | |
|----------|------------------------------------|-----------|
| 1 | Contenu | 3 |
| 1.1 | Démarrage rapide | 3 |
| 1.2 | Fichiers d'entrée | 4 |
| 1.3 | Fichiers de sortie | 5 |
| 1.4 | Interface de haut niveau | 6 |
| 1.5 | Référence de l'IPA | 7 |
| 1.6 | Contribuer | 15 |
| 1.7 | Remerciements | 16 |
| 2 | Index et tableaux | 17 |
| | Index des modules Python | 19 |
| | Index | 21 |

TraDSSAT vise à réinventer la roue, une fois pour toutes, en ce qui traite de la gestion des fichiers [DSSAT](#). Qu'il s'agisse de la préparation de fichiers d'entrées de DSSAT pour des simulations automatisées ou de l'analyse des résultats, l'interface simple en Python et la structure modulaire de traDSSAT simplifieront votre code de manière dramatique.

1.1 Démarrage rapide

1.1.1 Installation

traDSSAT nécessite **numpy** et **chardet**. Vous pouvez l'installer de **pip** avec : `pip install tradssat`
Vous pouvez aussi installer la version en développement la plus récente directement de **GitHub** avec : `pip install git+git://github.com/julienmalard/tradssat.git@master`

Vous aurez besoin d'une installation locale de **DSSAT** pour utiliser les gestionnaires de fichier de l'interface de haut niveau (parce que ceux-ci devront pouvoir localiser les fichiers de sol et météo, parmi autres, dans le dossier d'installation de **DSSAT**). **TraDSSAT** devrait pouvoir trouver votre installation de **DSSAT** par elle-même, mais si elle a besoin d'aide vous pouvez la spécifier avec :

```
from tradssat import set_dssat_dir
set_dssat_dir('C:/My/odd/path/to/DSSAT47')
```

1.1.2 Un exemple rapide

```
from tradssat import SoilFile, WTHFile, ExpFile, set_dssat_dir
from tradssat import GeneticMgr, CULFile, ECOFile

# Read, edit and write soil files
soil = SoilFile('path/to/my/file.SOL')

# Read and write weather files as well
wth = WTHFile('path/to/my/WTHR0001.WTH')

# ...and experiment files!
exp = ExpFile('path/to/my/experiment.EXP')
```

(suite sur la page suivante)

(suite de la page précédente)

```
# Access genetic coefficients by cultivar file or ecotype file
cul = CULFile('path/to/my/MZIXM.CUL')
eco = ECOFile('path/to/my/MZIXM.ECO')

cul.get_value('P1') # returns array of all varieties' P1
eco.get_value('TOPT') # returns array of all ecotypes' TOPT

# ...or automagically!
set_dssat_dir('C:/DSSAT47')
gen = GeneticMgr(crop='MZIXM', cult='PC0001')
gen.get_value('P1') # Returns P1 for MZIXM cultivar PC0001
gen.get_value('TOPT') # Returns ecotype variable TOPT for cultivar PC0001
```

1.2 Fichiers d'entrée

TraDSSAT est dotée de classes spécifiques pour lire et éditer les fichiers d'intrant de DSSAT. Chaque classe définit ses propres variables permises, autant que les paramètres de celles-ci. Tous les fichiers ont la même structure générale :

1. Section (nommée ; l'entête commence avec un * ou un \$ dans les fichiers de DSSAT)
2. Variables d'entête de section (facultatif)
3. Sous-section (numérotée ; l'entête commence avec un @ suivit des noms des variables dans un fichier DSSAT)
4. Variables de sous-section

Les classes utilisées pour lire les fichiers d'entrée de DSSAT sont :

- *SoilFile* (.SOL)
- *WTHFile* (.WTH, .WTG)
- *ExpFile* (.ccX)
- *CULFile* (.CUL)
- *ECOFile* (.ECO)

Vu que tous les fichiers d'entrée héritent de *InpFile*, la même interface s'applique pour la lecture, la modification et l'écriture de tous les fichiers d'entrée DSSAT.

1.2.1 Lecture de fichiers

Les fichiers d'entrée sont initialisés avec le chemin du fichier à lire.

```
from tradssat import WTHFile
wth = WTHFile('path/to/my/WTHR0001.WTH')
```

Les valeurs de toutes les variables peuvent ensuite être lues directement.

```
# Get solar radiation data time series
wth.get_value('SRAD')

# Conditions can also be set

# Only get data from 2 Jan, 2013.
wth.get_value('SRAD', cond={'DATE': '13002'})

# Only get data from dry days
wth.get_value('SRAD', cond={'RAIN': 0})
```

C'est aussi facile d'obtenir une liste de tous les noms de variable permis pour le type de fichier.


```
wth.variables()
```

1.2.2 Modification des fichiers

Vous pouvez changer les valeurs des variables, qu'il s'agisse du fichier en entier ou d'une section ou sous-section particulière, autant que selon une condition.

```
from tradssat import SoilFile
soil = SoilFile('path/to/my/file.SOL')

# Set all soils' SALB to 0.15
soil.set_value('SALB', 0.15)

# Only set soil IB00000002's SALB to 0.2
soil.set_value('SALB', 0.20, sect='IB00000002')

# Increase clay, but only for the first 5 cm of soil IB00000002
soil.set_value('SLCL', 0.50, sect='IB00000002', cond={'SLB': 5})
```

Vous pouvez aussi ajouter des rangées à des sous-sections spécifiques d'un fichier, ou bien enlever des rangées existantes. Les variables de sous-section non spécifiées dans `vals` seront assignée une valeur manquante (généralement -99).

```
# Add new soil layer
soil.add_row(
    sect='IB00000002', subsect=2, vals={'SLB': 180, 'SLLL': 0.260}
)
```

Vous pouvez sauvegarder les données en format json, ou bien écrire un fichier en format DSSAT sur le disque.

```
# Convert to dict...
json_d = soil.to_dict()

# ...or save to disk
wth.write('path/to/my/new/SOIL.SOL')
```

1.3 Fichiers de sortie

Les fichiers de sortie ne peuvent être que lus (et non écrits). Les classes suivantes sont disponibles :

- *PlantGroOut* (PlantGro.OUT)
- *SoilNiOut* (SoilNi.OUT)
- *SoilTempOut* (SoilTemp.OUT)
- *SummaryOut* (Summary.OUT)

Ces classes peuvent toutes être utilisées pour lire des fichiers de sortie correspondants. Cependant, veuillez noter qu'il est généralement plus facile et plaisant d'utiliser un objet de la classe *DSSATResults* au lieu d'accéder directement aux fichiers de sortie individuels.

```
from tradssat.out import PlantGroOut

plant = PlantGroOut('path/to/my/PlantGro.OUT')

# Get Leaf Area Index time series
plant.get_value('LAID')
```

1.4 Interface de haut niveau

L'interface d'haut niveau simplifie beaucoup de tâches communes lorsque vous travaillez avec des fichiers d'entrée et de sortie de DSSAT.

1.4.1 Fichiers d'entrée

Chacun de ces gestionnaires crée une interface plus intuitive pour les fichiers DSSAT que celle des lecteurs de fichier de base.

Gestionnaire de fichier d'expérience

DSSATRun vous permet d'éditer et d'écrire des expériences, et cela avec accès automatique aux fichiers externes de sol, génétique et météo associés. En plus, TraDSSAT gère automatiquement le lien entre des traitements de DSSAT et les niveaux de facteurs associés (si vous ne savez pas de quoi je parle, ce serait un bon moment d'arrêter de lire ceci et d'aller lire la documentation de DSSAT à la place).

```
from tradssat import DSSATRun

run = DSSATRun('path/to/my/experiment.EXP')

# Get cultivar for treatment 1
run.get_trt_val('INGENO', trt=1)

# Change level of treatment factor
run.set_trt_factor_level(trt=1, factor='CULTIVARS', level=2)

# Change value of a factor level (in this case cultivar type)
run.set_factor_level_val('INGENO', 'IB0067', level=1)

# Access soil variable SLLL for treatment 2
run.get_trt_val('SLLL', trt=2)
```

Gestionnaire de fichier de génétique

Les modules de culture de DSSAT séparent généralement les coefficients entre fichiers de cultivar, d'écotype et d'espèce. TraDSSAT offre la classe *GeneticMgr* pour automatiquement gérer tous les coefficients génétiques pour cultivar donné.

```
from tradssat import GeneticMgr

gen = GeneticMgr(crop='MZIXM', cult='PC0001')

# Returns P1 for MZIXM cultivar PC0001
gen.get_value('P1')

# Returns ecotype variable TOPT for cultivar PC001
gen.get_value('TOPT')
```

Note : Il est actuellement impossible d'accéder aux coefficients d'espèce avec traDSSAT, parce que ceux-ci sont en pratique des constants du modèle et ne devraient pas être modifiés ou changés (et, s'ils l'étaient, auraient aussi comme

effet d'affecter **toutes les simulations futures** de votre installation de DSSAT). D'un point de vue plus pratique, il y a aussi le fait que ces fichiers viennent dans une multitude de formats et seraient donc un drôle de casse-tête à analyser).

Gestionnaire de fichier de sol

Les fichiers de sol peuvent être difficiles à gérer, parce qu'ils peuvent contenir les données de beaucoup de différents sols dans le même fichiers. Avec la classe *SoilMgr*, vous n'avez qu'à spécifier le code du sol et traDSSAT trouvera le fichier et la section correspondants.

```
from tradssat.mgrs import SoilMgr

soil_mgr = SoilMgr('IB00000005')
soil_mgr.get_value('SLU1')
```

Gestionnaire de fichier de météo

Vous n'avez pas le goût de chercher vous-même pour le fichier de météo dont vous avez de besoin ? Vous n'avez qu'à donner le code de la station à *WeatherFileMgr* et elle la trouvera pour vous.

```
from tradssat.mgrs import WeatherFileMgr

wth_mgr = WeatherFileMgr('ACNM')
wth_mgr.get_value('RAIN')
```

1.4.2 Fichiers de sortie

Vous pouvez accéder les résultats d'une simulation à l'aide d'un objet *DSSATResults* initialisé avec le dossier contenant les résultats.

```
from tradssat import DSSATResults

out = DSSATResults('path/to/my/output/dir')

# Get FWAD results for treatment 1 (as a time series)
out.get_value('FWAD', trt=1)

# Get results at specific time

# Get result at 13 days after start
out.get_value('FWAD', trt=1, t=13, at='DAS')

# Get result at 13 days after planting
out.get_value('FWAD', trt=1, t=13, at='DAP')

# Get result at 123th day of year 1989
out.get_value('FWAD', trt=1, t='1989 123', at='DOY')
```

1.5 Référence de l'IPA

TraDSSAT permet aux utilisatrices et utilisateurs d'accéder les fichiers directement autant qu'à travers sont interface de haut niveau.

1.5.1 Gestionnaires de fichiers

Tout ce qui traite de lecture et de modification de fichiers DSSAT est effectué par des sous-classes de ces classes gabarit.

Fichiers d'entrée

Tous les fichiers d'entrée héritent de *InpFile*.

```
class tradssat.SoilFile (file)  
    Lecteur pour les fichiers de sol (.SOL) de DSSAT.  
  
class tradssat.WTHFile (file)  
    Lecteur pour les fichiers de météo (.WTH et .WTG) de DSSAT.  
  
class tradssat.ExpFile (file)  
    Lecteur pour les fichiers d'expérience (.ccX) de DSSAT.  
  
class tradssat.CULFile (file)  
    Lecteur de fichiers de cultivar (.CUL).  
  
class tradssat.ECOFile (file)  
    Lecteur de fichiers d'écotype (.ECO).
```

Fichiers de sortie

La majorité des fichiers de sortie héritent de *OutFile*.

```
class tradssat.out.PlantGroOut (folder)  
    Lecteur pour les fichiers de résultats de croissance de culture (PLANTGRO.OUT) de DSSAT.  
  
class tradssat.out.SoilNiOut (folder)  
    Lecteur pour les fichiers de résultats d'azote des sols (SOILNI.OUT) de DSSAT.  
  
class tradssat.out.SoilTempOut (folder)  
    Lecteur pour les fichiers de résultats de température des sols (SOILTEMP.OUT) de DSSAT.  
  
class tradssat.out.SummaryOut (folder)  
    Lecteur pour les fichiers de résumé de simulation (SUMMARY.OUT) de DSSAT.
```

1.5.2 IPA de haut niveau

L'IPA de haut niveau simplifie le travail associé avec la gestion des fichiers DSSAT plus complexes, surtout lorsqu'il y a des références vers d'autres fichiers.

Gestionnaires d'entrées de simulation

```
class tradssat.mgrs.DSSATRun (file, model=None)  
    Gestionnaire général pour les fichiers d'entrée de DSSAT.  
    add_factor_level (factor, vals)  
        Ajoute un facteur à l'expérience.  
        Paramètres  
        — factor (str) – Code ou nom du facteur.
```

- **vals** (*dict*) – Dictionnaire des valeurs des variables pour ce facteur. Les variables manquantes seront assignées le code correspondant pour les valeurs manquantes (généralement -99).

add_treatment (*name, ops=None, factors=None*)

Ajoute un niveau de traitement à l'expérience.

Paramètres

- **name** (*str*) – Le nom du nouveau traitement.
- **ops** (*dict*) – Dictionnaire des valeurs pour les options *R*, *O* et *C* du niveau de traitement (facultatif).
- **factors** (*dict*) – Dictionnaire des valeurs pour les niveaux des facteurs du traitement (facultatif). Les facteurs manquants seront assignés le niveau 0.

get_factor_level_val (*var, level*)

Obtenir la valeur d'une variable pour un niveau de facteur spécifique.

Paramètres

- **var** (*str*) – La variable d'intérêt.
- **level** (*int*) – Le niveau de facteur correspondant à la variable spécifiée.

Renvoie La valeur de la variable pour le niveau du facteur.

Type renvoyé np.ndarray

get_general_val (*var*)

Obtenir une valeur de variable de la section *GENERAL* du fichier EXP.

Paramètres **var** (*str*) – Le nom de la variable

Renvoie La valeur de la variable

Type renvoyé np.ndarray

get_trt_factor_level (*trt, factor*)

Obtenir le niveau de facteur pour un traitement donné.

Paramètres

- **trt** (*str* | *int*) – Nom ou numéro du traitement.
- **factor** (*str*) – Nom ou code du facteur.

Renvoie Le niveau de facteur correspondant au traitement donné.

Type renvoyé int

get_trt_name (*n*)

Retourne le nom du traitement correspondant au numéro de traitement donné.

Paramètres **n** (*int*) – Le numéro de traitement.

Renvoie Le nom du traitement.

Type renvoyé str

get_trt_num (*trt*)

Retourne le numéro de traitement correspondant au nom de traitement spécifié.

Paramètres **trt** (*str*) – Le nom du traitement.

Renvoie Le nom du traitement correspondant.

Type renvoyé int

get_trt_val (*var, trt*)

Retourne la valeur d'une variable du traitement.

Paramètres

- **var** (*str*) – La variable d'intérêt.
- **trt** (*int* | *str*) – Le nom ou numéro de traitement.

Renvoie La valeur de la variable pour le traitement.

Type renvoyé np.ndarray

remove_treatment (*trt*)

Efface un traitement de l'expérience.

Paramètres **trt** (*str* | *int*) – Nom ou numéro du traitement.

set_general_val (*var*, *val*)

Définit la valeur d'une variable dans la section *GENERAL* du fichier EXP.

Paramètres

— **var** (*str*) – Le nom de la variable.

— **val** (*str* | *float* | *int* | *np.ndarray*) – La nouvelle valeur pour la variable.

set_trt_factor_level (*trt*, *factor*, *level*)

Définit le niveau d'un facteur d'un traitement.

Paramètres

— **trt** (*str* | *int*) – Le nom ou numéro de traitement.

— **factor** (*str*) – Le nom ou code du facteur.

— **level** (*int*) – Le nouveau niveau du facteur.

treatments (*name=False*)

Retourne les traitements présents dans la simulation.

Paramètres **name** (*bool*) – Si vous voulez les noms ou les numéros des traitements.

Renvoie La liste des traitements.

Type renvoyé *np.ndarray*

class tradssat.mgrs.**GeneticMgr** (*crop*, *cult*)

class tradssat.mgrs.**SoilMgr** (*code*)

class tradssat.mgrs.**WeatherFileMgr** (*code*)

Gestionnaire des résultats de simulation

class tradssat.mgrs.**DSSATResults** (*folder*)

Facilite la lecture des résultats de DSSAT. Plutôt que de devoir lire chaque fichier de résultat un par un, vous n'avez qu'à donner à cette classe un dossier contenant les résultats d'une simulation DSSAT pour qu'elle y trouve les variables que vous voulez.

get_final_value (*var*, *trt*)

Retourne la valeur finale d'une variable. Plus rapide que *get_value()* si la variable est disponible dans *Summary.OUT*.

Paramètres

— **var** (*str*) – Le nom de la variable

— **trt** (*int*) – Le numéro du traitement d'intérêt.

Renvoie La valeur final de la variable.

Type renvoyé *str* | *float* | *int*

get_value (*var*, *trt*, *t=None*, *at='YEAR DOY'*)

Retourne la valeur (ponctuelle ou en série chronologique) d'une variable d'une simulation de DSSAT.

Paramètres

— **var** (*str*) – Le nom de la variable

— **trt** (*int*) – Le numéro du traitement d'intérêt.

— **t** (*str* | *int*) – Le temps duquel vous voulez les résultats. Si vous donnez *None*, les résultats de chaque point temporel seront retournés.

— **at** (*str*) – Doit être l'un de *DAS* (jours depuis le début), *DAP* (jours après la semence), ou *YEAR DOY* (année, jour de l'année; valeur par défaut). Seulement utilisé si *t* n'est pas *None*.

Renvoie La valeur de la variable

Type renvoyé np.ndarray

reload()

Recharger les données (util si une nouvelle simulation de DSSAT a été exécutée).

1.5.3 Fonctionnement interne de traDSSAT

Cette section document l'IPA (en majorité) interne de DSSAT et n'est utile que si vous planifiez vous-même contribuer ou apporter des modifications au code source.

Variables

Les types de variables utilisés pour spécifier les variables des fichiers DSSAT.

```
class tradssat.tmpl.var.CharacterVar (name, size, spc=1, sect=None, header_fill=' ', miss='-99', info="")
```

Variable texte (chaîne).

type_

alias de builtins.str

```
class tradssat.tmpl.var.FloatVar (name, size, dec, lims=None, spc=1, sect=None, header_fill=' ', miss='-99', info="")
```

Variable de virgule flottante.

type_

alias de builtins.float

```
class tradssat.tmpl.var.HeaderVariableSet (d_vars)
```

Organisateur pour les variables d'entête permises pour une classe de fichier DSSAT.

```
class tradssat.tmpl.var.IntegerVar (name, size, lims=None, spc=1, sect=None, header_fill=' ', miss='-99', info="")
```

Variable de chiffre entier.

type_

alias de builtins.int

```
class tradssat.tmpl.var.NumericVar (name, size, lims, spc, header_fill, miss, sect, info)
```

Classe mère pour les types de variable numériques.

```
class tradssat.tmpl.var.Variable (name, size, spc, header_fill, float_r, miss, sect, info)
```

Classe mère pour tous les types de variable.

```
class tradssat.tmpl.var.VariableSet (l_vars)
```

Organisateur pour les variables permises dans une classe de fichier DSSAT.

Valeurs

La structure interne utilisée pour organiser les données de fichier DSSAT et les valeurs de leurs variables.

```
class tradssat.tmpl.vals.FileValueSet
```

Représente l'ensemble des valeurs dans un fichier DSSAT.

```
add_row (sect, subsect=None, vals=None)
```

Ajoute un rangée au fichier.

Paramètres

- **sect** (*str*) – Nom de la section.
- **subsect** (*int*) – Numéro de la sous-section. Si `None`, un rangée sera ajoutée à toutes les sous-sections.

— **vals** (*dict*) – Dictionnaire des valeurs de variables pour la nouvelle rangée.

add_section (*name*)

Ajoute une section au fichier.

Paramètres **name** (*str*) – Nom de la nouvelle section.

changed ()

Détecte si les valeurs des variables ont changées.

Renvoie

Type renvoyé bool

find_var_sect (*var*)

Trouve la section dans laquelle apparaît une variable

Paramètres **var** (*str*) – Le nom de la variable

Renvoie Le nom de la section du fichier.

Type renvoyé str

to_dict ()

Convertie le fichier à un dictionnaire.

Renvoie

Type renvoyé dict

write (*lines*)

Écrit le fichier.

Paramètres **lines** (*list[str]*) – La liste à laquelle les lignes de sortie seront écrites.

Renvoie La liste modifiée.

Type renvoyé list

class tradssat.tmpl.vals.**HeaderValues**

Représente les variables d’entête de fichier DSSAT et leurs valeurs.

changed ()

Vérifie si les valeurs des variables d’entête ont été changées.

Renvoie

Type renvoyé bool

get_value (*var*)

Obtenir la valeur d’une variable d’entête.

Paramètres **var** (*str*) – La variable d’intérêt.

Renvoie La valeur de la variable

Type renvoyé np.ndarray

set_vars (*subsect*)

Définie les variables de l’entête.

Paramètres **subsect** (*ValueSubSection*) – La sous-section avec ses variables et leurs valeurs déjà spécifiées.

to_dict ()

Convertir au format de dictionnaire (json).

Renvoie Les variables de l’entête en format (pas mal) jsonifié.

Type renvoyé list

write ()

Écrit les valeurs de l’entête vers une valeur de chaîne.

Renvoie

Type renvoyé str

```

class tradssat.tmpl.vals.ValueSection (name)
    Représente la structure et les valeurs des variables d’une section d’un fichier de DSSAT.
    add_subsection (subsect)
        Ajoute une sous-section à la section.
        Paramètres subsect (ValueSubSection) – La nouvelle sous-section.
    changed ()
        Vérifie si une valeur a été changée dans la section.
        Renvoie
        Type renvoyé bool
    get_header_var (var)
        Obtenir la valeur d’une variable d’entête.
        Paramètres var (str) – Le nom de la variable
        Renvoie La valeur de la variable d’entête.
        Type renvoyé np.ndarray
    set_header_vars (h_vars)
        Définit les variables de l’entête de la section.
        Paramètres h_vars (ValueSubSection) – La sous-section représentant les variables d’en-
            tête.
    to_dict ()
        Convertit la section en dictionnaire.
        Renvoie
        Type renvoyé dict
    write (lines)
        Écrit la section en liste de lignes.
        Paramètres lines (list[str]) – La liste de lignes à laquelle ajouter cette section.

class tradssat.tmpl.vals.ValueSubSection (l_vars, l_vals)
    Représente les variables d’une sous-section d’un fichier DSSAT avec leurs valeurs.
    add_row (vals=None)
        Ajoute une rangée à la sous-section.
        Paramètres vals (dict) – Les valeurs pour la nouvelle rangée. Si un valeur manque, elle sera
            assigné le code de valeur manquante correspondant (généralement -99).
    changed ()
        Vérifie si une variable dans la sous-section a changé de valeur.
        Renvoie
        Type renvoyé bool
    check_dims ()
        Vérifie que toutes les variables dans la sous-section ont la même taille (sinon, la sous-section ne pourra
            être sauvegarder sur le disque).
        soulève ValueError – Si toutes les variables n’ont pas la même taille.
    n_data ()
        Obtenir la taille des variables. Soulèvera une erreur si toutes les variables n’ont pas la même taille.
        Renvoie La taille de toutes les variables de la sous-section.
        Type renvoyé int
        soulève ValueError – Si toutes les variables n’ont pas la même taille.
    to_dict ()
        Convertie la sous-section en dictionnaire.

```

Renvoie**Type renvoyé** dict**write** (*lines*)

Écrit la sous-section.

Paramètres **lines** (*list[str]*) – La liste de lignes à laquelle ajouter cette sous-section.**class** tradssat.tmpl.vals.**VariableValue** (*var, val*)

Représente une variable de fichier DSSAT.

add_value (*val*)

Ajoute une valeur à la matrice de la variable.

Paramètres **val** (*int | float*) – La nouvelle valeur.**remove_value** (*i*)

Enlève une valeur de la matrice de la variable.

Paramètres **i** (*np.ndarray*) – Les indices des valeur(s) à effacer. Cela peut être une liste d’indices, ou bien un masque booléen de la même taille que la variable.**set_value** (*val, i=None*)

Modifie la valeur de la variable

Paramètres— **val** (*float | int | np.ndarray*) – La nouvelle valeur.— **i** (*int | np.ndarray*) – Les indices auxquels changer la valeur. Si *None*, toutes les valeurs de la variable seront changées.**size** ()

Retourne la taille de la variable.

Renvoie**Type renvoyé** int**write** (*i=None*)

Convertie la variable en chaîne.

Paramètres **i** (*int*) – L’indice de la valeur à écrire. Si *None*, le nom de la variable sera écrit à la place.**Renvoie** La variable, correctement formatée et écrite en format de chaîne.**Type renvoyé** str

Fichiers d’entrée et de résultats

Tous les objets de fichiers, qu’ils soient d’entrée ou de résultats, héritent de la classe abstraite *File* et de toutes ses méthodes.

class tradssat.tmpl.**InpFile** (*file*)Classe mère pour tous les fichiers d’entrée, autant que pour *Summary.OUT*.**changed** ()

Vérifie si le fichier a été édité et doit être réécrit.

Renvoie Si le fichier a été édité ou non.**Type renvoyé** bool**classmethod** **matches_file** (*file*)

Vérifie si un fichier donné peut être lu par cette classe.

Paramètres **file** (*str*) – Le nom du fichier à lire, ou son chemin complet.**Renvoie** *True* si le fichier correspond ; *False* sinon.**Type renvoyé** bool

```
class tradssat.tmpl.OutFile (folder)
    Classe mère pour (quasiment) tous les fichiers de résultat DSSAT.
classmethod matches_file (file)
    Vérifie si un fichier donné peut être lu par cette classe.
        Paramètres file (str) – Le nom du fichier à lire, ou son chemin complet.
        Renvoie True si le fichier correspond ; False sinon.
        Type renvoyé bool
```

Gabarit de fichier

Tous les objets de fichiers, qu'ils soient d'entrée ou de résultats, héritent de la classe abstraite *File* et de toutes ses méthodes.

```
class tradssat.tmpl.File (file)
    Classe mère pour tous les objets fichier.
get_value (var, sect=None, subsect=None, cond=None)
    Paramètres
        — var –
        — sect –
        — subsect –
        — cond –
    Renvoie
        Type renvoyé np.ndarray
get_var_size (var, sect=None)
    Retourne la taille d'une variable.
    Paramètres
        — var (str) – Le nom de la variable.
        — sect (str) – Le nom de la section dans laquelle cette variable apparaît (facultatif ; utilisé
            pour les cas ambigus ou un fichier contient plusieurs variables avec le même code).
    Renvoie La taille de la variable
    Type renvoyé int
classmethod matches_file (file)
    Vérifie si un fichier donné peut être lu par cette classe. Doit être implémenté dans les sous-classes.
        Paramètres file (str) – Le fichier à lire.
        Renvoie True si le fichier correspond ; False sinon.
        Type renvoyé bool
```

1.6 Contribuer

Il y a plusieurs méthodes de contribuer à traDSSAT...

1.6.1 Signalisation d'erreurs

Quelque chose paraît bizarre ? SVP signalez le sur [GitHub](#).

1.6.2 Améliorations

Vous pensez pouvoir le réparer vous-même ? Merveilleux ! Bifurquez traDSSAT sur [GitHub](#) et soumettez une requête de téléchargement avec vos modifications.

1.7 Remerciements

1.7.1 Auteurs

TraDSSAT est développé par :

- [Julien Jean Malard](#)
- (Shreya Yadav)

CHAPITRE 2

Index et tableaux

- `genindex`
- `search`

t

`tradssat.tmpl.vals`, [11](#)

`tradssat.tmpl.var`, [11](#)

A

add_factor_level() (méthode trad-
ssat.mgrs.DSSATRun), 8

add_row() (méthode tradssat.templ.vals.FileValueSet),
11

add_row() (méthode trad-
ssat.templ.vals.ValueSubSection), 13

add_section() (méthode trad-
ssat.templ.vals.FileValueSet), 12

add_subsection() (méthode trad-
ssat.templ.vals.ValueSection), 13

add_treatment() (méthode trad-
ssat.mgrs.DSSATRun), 9

add_value() (méthode trad-
ssat.templ.vals.VariableValue), 14

C

changed() (méthode tradssat.templ.InpFile), 14

changed() (méthode tradssat.templ.vals.FileValueSet),
12

changed() (méthode trad-
ssat.templ.vals.HeaderValues), 12

changed() (méthode tradssat.templ.vals.ValueSection),
13

changed() (méthode trad-
ssat.templ.vals.ValueSubSection), 13

CharacterVar (classe dans tradssat.templ.var), 11

check_dims() (méthode trad-
ssat.templ.vals.ValueSubSection), 13

CULFile (classe dans tradssat), 8

D

DSSATResults (classe dans tradssat.mgrs), 10

DSSATRun (classe dans tradssat.mgrs), 8

E

ECOFile (classe dans tradssat), 8

ExpFile (classe dans tradssat), 8

F

File (classe dans tradssat.templ), 15

FileValueSet (classe dans tradssat.templ.vals), 11

find_var_sect() (méthode trad-
ssat.templ.vals.FileValueSet), 12

FloatVar (classe dans tradssat.templ.var), 11

G

GeneticMgr (classe dans tradssat.mgrs), 10

get_factor_level_val() (méthode trad-
ssat.mgrs.DSSATRun), 9

get_final_value() (méthode trad-
ssat.mgrs.DSSATResults), 10

get_general_val() (méthode trad-
ssat.mgrs.DSSATRun), 9

get_header_var() (méthode trad-
ssat.templ.vals.ValueSection), 13

get_trt_factor_level() (méthode trad-
ssat.mgrs.DSSATRun), 9

get_trt_name() (méthode trad-
ssat.mgrs.DSSATRun), 9

get_trt_num() (méthode tradssat.mgrs.DSSATRun),
9

get_trt_val() (méthode tradssat.mgrs.DSSATRun),
9

get_value() (méthode tradssat.mgrs.DSSATResults),
10

get_value() (méthode tradssat.templ.File), 15

get_value() (méthode trad-
ssat.templ.vals.HeaderValues), 12

get_var_size() (méthode tradssat.templ.File), 15

H

HeaderValues (classe dans tradssat.templ.vals), 12

HeaderVariableSet (classe dans tradssat.templ.var),
11

I

InpFile (classe dans tradssat.templ), 14

IntegerVar (classe dans tradssat.tmpl.var), 11

M

matches_file() (méthode de la classe tradssat.tmpl.File), 15

matches_file() (méthode de la classe tradssat.tmpl.InpFile), 14

matches_file() (méthode de la classe tradssat.tmpl.OutFile), 15

N

n_data() (méthode tradssat.tmpl.vals.ValueSubSection), 13

NumericVar (classe dans tradssat.tmpl.var), 11

O

OutFile (classe dans tradssat.tmpl), 14

P

PlantGroOut (classe dans tradssat.out), 8

R

reload() (méthode tradssat.mgrs.DSSATResults), 11

remove_treatment() (méthode tradssat.mgrs.DSSATRun), 9

remove_value() (méthode tradssat.tmpl.vals.VariableValue), 14

S

set_general_val() (méthode tradssat.mgrs.DSSATRun), 10

set_header_vars() (méthode tradssat.tmpl.vals.ValueSection), 13

set_trt_factor_level() (méthode tradssat.mgrs.DSSATRun), 10

set_value() (méthode tradssat.tmpl.vals.VariableValue), 14

set_vars() (méthode tradssat.tmpl.vals.HeaderValues), 12

size() (méthode tradssat.tmpl.vals.VariableValue), 14

SoilFile (classe dans tradssat), 8

SoilMgr (classe dans tradssat.mgrs), 10

SoilNiOut (classe dans tradssat.out), 8

SoilTempOut (classe dans tradssat.out), 8

SummaryOut (classe dans tradssat.out), 8

T

to_dict() (méthode tradssat.tmpl.vals.FileValueSet), 12

to_dict() (méthode tradssat.tmpl.vals.HeaderValues), 12

to_dict() (méthode tradssat.tmpl.vals.ValueSection), 13

to_dict() (méthode tradssat.tmpl.vals.ValueSubSection), 13

tradssat.tmpl.vals (module), 11

tradssat.tmpl.var (module), 11

treatments() (méthode tradssat.mgrs.DSSATRun), 10

type_ (attribut tradssat.tmpl.var.CharacterVar), 11

type_ (attribut tradssat.tmpl.var.FloatVar), 11

type_ (attribut tradssat.tmpl.var.IntegerVar), 11

V

ValueSection (classe dans tradssat.tmpl.vals), 12

ValueSubSection (classe dans tradssat.tmpl.vals), 13

Variable (classe dans tradssat.tmpl.var), 11

VariableSet (classe dans tradssat.tmpl.var), 11

VariableValue (classe dans tradssat.tmpl.vals), 14

W

WeatherFileMgr (classe dans tradssat.mgrs), 10

write() (méthode tradssat.tmpl.vals.FileValueSet), 12

write() (méthode tradssat.tmpl.vals.HeaderValues), 12

write() (méthode tradssat.tmpl.vals.ValueSection), 13

write() (méthode tradssat.tmpl.vals.ValueSubSection), 14

write() (méthode tradssat.tmpl.vals.VariableValue), 14

WTHFile (classe dans tradssat), 8