



Trends.Earth - Guide du développeur

Version 2.1.8

Conservation International

févr. 06, 2023

1	Développement	2
1.1	Modifier le code du module d'extension QGIS	2
1.2	Modification du code de traitement de Earth Engine	6
1.3	Modification des modèles de couches vectorielles	8
1.4	Traitement des métadonnées des ensembles de données	10
1.5	Mise à jour du cadre de présentation des rapports	10
1.6	Contribuer à la documentation	14
2	Journal des modifications	18
2.1	2.1.8 (January 25, 2023)	18
2.2	2.1.6 (December 8, 2022)	18
2.3	2.1.6 (November 21, 2022)	18
2.4	2.1.2 (November 18, 2022)	19
2.5	2.1.0 (November 11, 2022)	19
2.6	2.0.7 (October 31, 2022)	19
2.7	2.0.5 (October 19, 2022)	19
2.8	2.0.3 (October 19, 2022)	19
2.9	2.0.1 (October 13, 2022)	19
2.10	2.0 (July 20, 2022)	20
2.11	1.0.10 (7 juillet 2022)	20
2.12	1.0.8 (15 octobre 2021)	20
2.13	1.0.6 (15 juillet 2021)	20
2.14	1.0.4 (30 juin 2021)	20
2.15	1.0.2 (14 août 2020)	21
2.16	1.0.0 (27 avril 2020)	21
2.17	0.98 (2 avril 2020)	22
2.18	0.66 (20 juillet 2019)	22
2.19	0.64 (9 juillet 2019)	23
2.20	0.62 (27 janvier 2019)	23
2.21	0.60 (3 décembre 2018)	23
2.22	0.58 (11 août 2018)	23
2.23	0.56.5 (21 mai 2018)	24
2.24	0.56.4 (21 mai 2018)	24
2.25	0.56.3 (21 avril 2018)	24
2.26	0.56.2 (10 avril 2018)	24
2.27	0.56.1 (10 avril 2018)	24

2.28	0.56 (9 avril 2018)	24
2.29	0.54 (8 avril 2018)	25
2.30	0.52.1 (21 mars 2018)	25
2.31	0.52.1 (21 mars 2018)	25
2.32	0.52 (19 mars 2018)	25
2.33	0.50 (15 mars 2018)	25
2.34	0.48 (13 mars 2018)	25
2.35	0.46 (13 mars 2018)	25
2.36	0.44 (12 mars 2018)	26
2.37	0.42 (4 février 2018)	26
2.38	0.40 (4 février 2018)	26
2.39	0.38 (16 janvier 2018)	27
2.40	0.36 (14 décembre 2017)	27
2.41	0.34 (14 décembre 2017)	28
2.42	0.32 (14 décembre 2017)	28
2.43	0.30 (12 décembre 2017)	28
2.44	0.24 (6 décembre 2017) < https://github.com/ConservationInternational/trends.earth/releases/tag/0.24 >`_	28
2.45	0.22 (4 décembre 2017) < https://github.com/ConservationInternational/trends.earth/releases/tag/0.22 >`_	28
2.46	0.18 (2 décembre 2017)	28
2.47	0.16 (6 novembre 2017)	28
2.48	0.14 (25 octobre 2017) < https://github.com/ConservationInternational/trends.earth/releases/tag/0.14 >`_	28
2.49	0.12 (6 octobre 2017)	28

▲TRENDS.EARTH est un logiciel libre et open-source, sous la [licence publique générale GNU](#), version 2.0 ou ultérieure.

L'outil ▲TRENDS.EARTH comporte un certain nombre de composants. Le premier est un module d'extension QGIS qui prend en charge le calcul des indicateurs, l'accès aux données brutes, la création de rapports et la production de cartes imprimées. Le code du module d'extension, ainsi que des instructions supplémentaires sur son installation si vous souhaitez modifier le code, se trouvent dans le dépôt GitHub de [trends.earth](#).

Le module d'extension QGIS ▲TRENDS.EARTH est soutenu par un certain nombre de scripts Python différents qui permettent de calculer les divers indicateurs sur Google Earth Engine (GEE). Ces scripts se trouvent dans le sous-dossier « gee » de ce dépôt GitHub. Les scripts GEE sont pris en charge par le module Python *landdegradation*, qui comprend le code pour le traitement des entrées et des sorties pour le module d'extension, ainsi que d'autres fonctions communes prenant en charge le calcul des intégrales NDVI, la signification statistique et d'autres codes partagés. Le code de ce module est disponible dans le dépôt [landdegradation](#) sur GitHub.

Vous trouverez ci-dessous de plus amples détails sur la façon de contribuer à Trends.Earth en travaillant sur le code du module d'extension, en modifiant le code de traitement, ou en contribuant à la traduction du site Web et du module d'extension.

1.1 Modifier le code du module d'extension QGIS

1.1.1 Télécharger le code de trends.earth

Le code Trends.Earth, tant pour le module d'extension que pour les scripts Google Earth Engine qui le prennent en charge, se trouve sur GitHub dans le dépôt [trends.earth](#). Clonez ce dépôt à un endroit approprié sur votre machine afin de vous assurer que vous disposez de la dernière version du code.

Il existe un certain nombre de sections différentes du dépôt trends.earth qui sont en cours de développement. Le module d'extension prend officiellement en charge QGIS3 et la majorité du développement a lieu sur la section « développer ». La section « qgis2 » est la version la plus ancienne du module d'extension, et prend en charge QGIS2 version 2.18+.

1.1.2 Installation des dépendances

Python

Le module d'extension est codé en Python. En plus d'être utilisé pour exécuter le module d'extension via QGIS, Python est également utilisé pour prendre en charge la gestion du module d'extension (changement de version, installation de versions de développement, etc.) Bien que Python soit inclus dans QGIS, vous aurez également besoin d'une version locale de Python que vous pourrez configurer avec le logiciel nécessaire à la gestion du module d'extension. Le moyen le plus simple de gérer plusieurs versions de Python est de passer par la « distribution Anaconda <<https://www.anaconda.com>> »_. Pour le travail de développement du module d'extension, Python 3 est nécessaire. Pour télécharger Python 3.7 (recommandé) via Anaconda, « consultez cette page <<https://www.anaconda.com/distribution/#download-section>> »_.

Dépendances Python

Afin de travailler avec le code de trends.earth, vous devez avoir Invoke installé sur votre appareil, ainsi qu'un certain nombre d'autres paquets qui sont utilisés pour gérer la documentation, les traductions, etc. Ces paquets sont tous listés dans le fichier d'exigences « dev » pour Trends.Earth. Ils peuvent donc être installés en naviguant dans une invite de commande à la racine du dossier du code trends.earth et en tapant

```
pip install -r requirements-dev.txt
```

Note: Si vous utilisez Anaconda, vous devrez d'abord activer un environnement virtuel Python 3.7 avant d'exécuter la commande ci-dessus (et toute autre commande invoke listée sur la page). Une façon de le faire est de lancer une « invite Anaconda », en suivant les instructions de cette page [Anaconda](#).

PyQt

PyQt5 est la boîte à outils graphique utilisée par QGIS3. Pour compiler l'interface utilisateur de Trends.Earth pour QGIS3, vous devez installer PyQt5. Ce paquet peut être installé à partir de pip en utilisant

```
pip install PyQt5
```

Note: PyQt4 est la boîte à outils graphique utilisée par QGIS2. La meilleure source pour ce paquetage sous Windows est l'ensemble de paquets maintenu par Christoph Gohlke à UC Irvine. Pour télécharger PyQt4, sélectionnez « le paquetage approprié sur cette page <<https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyqt4>> »_. Choisissez le fichier approprié pour la version de Python que vous utilisez. Par exemple, si vous utilisez Python 2.7, choisissez la version avec « cp27 » dans le nom du fichier. Si vous utilisez Python 3.7, choisissez la version dont le nom de fichier contient « cp37 ». Choisissez « amd64 » pour un python 64 bits, et « win32 » pour un python 32 bits.

Après avoir téléchargé depuis le lien ci-dessus, utilisez « pip » pour l'installer. Par exemple, pour la roue 64 bits pour Python 3.7, vous devez exécuter

```
pip install PyQt4-4.11.4-cp37-cp37m-win_amd64.whl
```

1.1.3 Changement de la version du module d'extension

La convention pour Trends.Earth est que les numéros de version se terminant par un nombre impair (par exemple 0,65) sont des versions de développement, tandis que les versions se terminant par un nombre pair (par exemple 0,66) sont des versions de lancement. Les versions de développement du module d'extension ne sont jamais publiées via le référentiel QGIS, et ne sont donc jamais vues par les utilisateurs normaux du module d'extension. Les versions de développement portant un numéro impair sont utilisées par l'équipe de développement de Trends.Earth pour tester les nouvelles fonctionnalités avant leur lancement public.

Si vous souhaitez apporter des modifications au code et que vous avez téléchargé une version publique du module d'extension (se terminant par un numéro pair), la première étape consiste à mettre à jour la version du module d'extension au numéro impair suivant. Ainsi, par exemple, si vous avez téléchargé la version 0,66 du module d'extension, vous devez mettre à jour la version 0,67 avant de commencer à effectuer vos modifications. Il y a plusieurs endroits dans le code où la version est mentionnée (ainsi que dans chaque script GEE), il y a donc une tâche invoke pour aider à changer la version. Pour que la version soit 0,67, vous devez exécuter

```
invoke set-version -v 0.67
```

L'exécution de la commande ci-dessus mettra à jour le numéro de version à chaque endroit où il est référencé dans le code. Pour éviter toute confusion, ne remplacez jamais la version par une version qui a déjà été publiée ; AUGMENTEZ toujours la valeur de la balise de version jusqu'au prochain nombre impair.

1.1.4 Test des modifications apportées au module d'extension

Après avoir apporté des modifications au code du module d'extension, vous devez les tester pour vous assurer que le module se comporte comme prévu et qu'aucun bogue ou erreur ne survient. Le module d'extension doit subir des tests approfondis avant d'être publié dans le référentiel QGIS (où d'autres utilisateurs peuvent y accéder) afin de s'assurer qu'aucune modification du code ne casse le module d'extension.

Pour tester les modifications que vous avez apportées au module d'extension dans QGIS, vous devez l'installer au niveau local. Il existe des tâches invoke qui vous aident dans ce processus. La première étape avant d'installer le module d'extension est de s'assurer que vous avez configuré le module avec toutes les dépendances dont il a besoin pour fonctionner à partir de QGIS. Pour ce faire, exécutez

```
invoke plugin-setup
```

La tâche ci-dessus ne doit être exécutée qu'immédiatement après le téléchargement du code de trends.earth, ou si des modifications sont apportées aux dépendances du module d'extension. Par défaut, « plugin-setup » réutilisera tous les fichiers mis en cache sur votre appareil. Pour commencer à partir de zéro, ajoutez le drapeau « -c » (clean) à la commande ci-dessus.

Après avoir exécuté « plugin-setup », vous êtes prêt à installer le module d'extension dans le dossier des modules d'extension QGIS sur votre machine. Pour ce faire, exécutez

```
invoke plugin-install
```

Après avoir exécuté la commande ci-dessus, vous devrez soit 1) redémarrer QGIS, soit 2) utiliser le « Plugin Reloader <https://plugins.qgis.org/plugins/plugin_reloader/> » pour recharger le module d'extension Trends.Earth afin de voir les effets des modifications que vous avez apportées.

Par défaut, « plugin-install » écrasera tout fichier de module d'extension existant sur votre machine, mais laissera en place toutes les données (limites administratives, etc.) que le module d'extension aurait pu télécharger. Pour repartir de zéro, ajoutez le drapeau « -c » (clean) à la commande ci-dessus. Il se peut que vous deviez fermer QGIS pour réussir une installation propre du module d'extension en utilisant le drapeau « -c ».

Note: Par défaut, « plugin-install » suppose que vous souhaitez installer le module d’extension à utiliser dans QGIS3. Pour installer le module d’extension pour l’utiliser dans QGIS3, ajoutez le drapeau « -v 2 » à la commande « plugin-install ». Rappelez-vous que le module d’extension peut ou non être entièrement fonctionnel sur QGIS3 ; le module d’extension a été conçu à l’origine pour QGIS2 et est toujours en cours de test sur QGIS3.

1.1.5 Synchronisation et déploiement des modifications apportées aux binaires

Pour accélérer les calculs dans Trends.Earth, certains outils permettent d’utiliser des binaires précompilés qui ont été compilés avec « numba <<https://numba.pydata.org>> ». Numba est un compilateur à source ouverte qui peut compiler le code Python et NumPy, le rendant plus rapide que lorsqu’il est exécuté en Python ordinaire. Pour éviter que les utilisateurs de Trends.Earth aient à télécharger Numba et toutes ses dépendances, l’équipe de Trends.Earth met à disposition des binaires pré-compilés à télécharger si les utilisateurs choisissent de les installer.

Pour générer des binaires précompilés pour le système d’exploitation, le bitness (32/64 bit) et la version de Python que vous exécutez sur votre machine, utilisez

```
invoke binaries-compile
```

Note: Vous aurez besoin d’un compilateur C++ pour que la commande ci-dessus fonctionne. Sous Windows, consultez « cette page github <https://wiki.python.org/moin/WindowsCompilers#Which_Microsoft_Visual_C.2B-.2B-_compiler_to_use_with_a_specific_Python_version_.3F> » pour savoir comment installer le compilateur Microsoft Visual C++ nécessaire à votre version de Python. Sous MacOS, vous devrez très probablement installer Xcode. Sur Linux, installez la version appropriée de GCC.

Pour rendre les binaires accessibles au public, ils sont distribués par le biais d’un seau S3 d’Amazon Web services. Pour télécharger les binaires générés avec la commande ci-dessus vers le seau, exécutez

```
invoke binaries-sync
```

Note: La commande ci-dessus échouera si vous n’avez pas de clés permettant l’accès en écriture au seau « trends.earth » sur S3.

La commande ci-dessus synchronisera chaque fichier binaire individuel vers S3. Cependant, les utilisateurs de la boîte à outils téléchargent les binaires sous forme d’un seul fichier ZIP lié à la version du module d’extension qu’ils utilisent. Pour générer ce fichier ZIP afin qu’il soit accessible aux utilisateurs de Trends.Earth, exécutez

```
invoke binaries-deploy
```

Note: La commande ci-dessus échouera si vous n’avez pas de clés permettant l’accès en écriture au seau « trends.earth » sur S3.

1.1.6 Création d'un fichier ZIP du module d'extension

Il existe plusieurs tâches `invoke` pour aider à créer un fichier ZIP afin de déployer le module d'extension dans le référentiel QGIS, ou pour partager la version de développement du module d'extension avec d'autres personnes. Pour emballer le module d'extension et toutes ses dépendances dans un fichier ZIP qui peut être installé en suivant « le processus décrit dans le fichier lisez-moi de Trends.Earth <<https://github.com/ConservationInternational/trends.earth#installing-latest-packaged-development-version>> », exécutez

```
invoke zipfile-build
```

Cette commande créera un dossier nommé « build » à la racine du dossier du code `trends.earth`, et dans ce dossier il créera un fichier appelé « LDMP.zip ». Ce fichier peut être partagé avec d'autres personnes, qui peuvent l'utiliser pour « installer manuellement Trends.Earth <<https://github.com/ConservationInternational/trends.earth#installing-latest-packaged-development-version>> ». Cette fonctionnalité peut être utile s'il y a un besoin de partager les dernières fonctionnalités avec quelqu'un avant qu'elles ne soient disponibles dans la version publique du module d'extension.

1.1.7 Déploiement du fichier ZIP de la version de développement

La page GitHub de Trends.Earth donne un lien vers un fichier ZIP qui permet aux utilisateurs qui ne sont pas forcément des développeurs d'accéder à la version de développement de Trends.Earth. Pour créer un fichier ZIP et le rendre disponible sur cette page (le fichier ZIP est stocké sur S3), exécutez

```
invoke zipfile-deploy
```

Cette commande va emballer le module d'extension et le copier sur <https://s3.amazonaws.com/trends.earth/sharing/LDMP.zip>.

Note: La commande ci-dessus échouera si vous n'avez pas de clés permettant l'accès en écriture au seau « `trends.earth` » sur S3.

1.2 Modification du code de traitement de Earth Engine

Les scripts de traitement Google Earth Engine (GEE) utilisés par Trends.Earth sont tous stockés dans le dossier « `gee` » sous le dossier principal de `trends.earth`. Pour que ces scripts soient accessibles aux utilisateurs du module d'extension QGIS de `trends.earth`, ils doivent être déployés sur le service `api.trends.earth` que Conservation International maintient afin de permettre aux utilisateurs du module d'extension d'utiliser Earth Engine sans avoir besoin de savoir programmer, ou d'avoir des comptes utilisateurs individuels sur GEE. La section ci-dessous décrit comment tester et déployer les scripts GEE à utiliser avec Trends.Earth.

1.2.1 Mise en place des dépendances

trends.earth-CLI

Le paquet Python « trends.earth-CLI » est nécessaire afin de travailler avec le serveur api.trends.earth. Ce paquet est situé sur GitHub dans le dépôt [trends.earth-CLI](#).

La première étape consiste à cloner ce dépôt sur votre appareil. Nous vous recommandons de cloner le dépôt dans le même dossier que celui où vous avez le code trends.earth. Par exemple, si vous avez un dossier « Code » sur votre appareil, clonez le dépôt [trends.earth](#) (le code pour le module d'extension QGIS et les scripts GEE associés) et aussi le dépôt [trends.earth-CLI](#) dans ce même dossier.

Lorsque vous configurez votre système comme recommandé ci-dessus, trends.earth-CLI fonctionnera avec les tâches invoke utilisées pour gérer trends.earth sans aucune modification. Si, cependant, vous téléchargez trends.earth-CLI dans un dossier différent, alors vous devrez ajouter un fichier nommé « invoke.yaml » à la racine du référentiel trends.earth, et dans ce fichier indiquer à Trends.Earth où localiser le code trends.earth-CLI. Ce fichier YAML devrait ressembler à ce qui suit (si vous avez téléchargé le code sous Windows dans un dossier appelé « C:/Users/azvol/Code/trends.earth-CLI/tecli ») :

```
gee:
  tecli: "C:/Users/azvol/Code/trends.earth-CLI/tecli"
```

Encore une fois, vous n'avez **pas** besoin d'ajouter ce fichier .yaml si vous configurez votre système comme recommandé ci-dessus.

docker

Le paquet trends.earth-CLI nécessite [docker](#) pour fonctionner. Suivez ces instructions pour installer docker sur Windows <<https://docs.docker.com/docker-for-windows/install/>>, et ces instructions pour installer docker sur macOS. Si vous utilisez Linux, suivez les instructions de cette page qui sont appropriées pour la distribution Linux que vous utilisez.

1.2.2 Tester localement un script Earth Engine

Après avoir installé le paquet trends.earth-CLI, vous devrez configurer un fichier .tecli.yml avec un jeton d'accès à un compte de service GEE afin de tester les scripts sur GEE. Pour configurer le compte de service GEE pour tecli, obtenez d'abord la clé de votre compte de service au format JSON (depuis la console Google cloud), puis encodez-la en base64. Fournissez cette clé encodée en base64 à tecli avec la commande suivante :

```
invoke tecli-config set EE_SERVICE_ACCOUNT_JSON key
```

où « clé » est la clé du compte de service au format JSON, codée en base64.

Lors de la conversion d'un script spécifiant le code à exécuter sur GEE de JavaScript à Python, ou lors de modifications de ce code, il peut être utile de tester le script localement, sans le déployer sur le serveur api.trends.earth. Pour ce faire, utilisez la tâche d'invocation « run ». Par exemple, pour tester le script « land_cover », allez dans le répertoire racine du code de Trends.Earth, et, dans une invite de commande, exécutez :

```
invoke tecli-run land_cover
```

Ceci utilisera le paquet trends.earth-CLI pour construire et exécuter un conteneur docker qui tentera d'exécuter le script « land_cover ». S'il y a des erreurs de syntaxe dans le script, elles apparaîtront lors de l'exécution du conteneur. Avant de soumettre un nouveau script à api.trends.earth, assurez-vous toujours que « invoke tecli-run » est capable d'exécuter le script sans aucune erreur.

Lorsque vous utilisez « invoke tecli-run », vous pouvez obtenir une erreur disant :

```
Invalid JWT: Token must be a short-lived token (60 minutes) and in a
reasonable timeframe. Check your iat and exp values and use a clock with
skew to account for clock differences between systems.
```

Cette erreur peut être causée si l'horloge du conteneur docker n'est pas synchronisée avec l'horloge du système. Le redémarrage de Docker devrait corriger cette erreur.

1.2.3 Déploiement d'un script GEE sur api.trends.earth

Lorsque vous avez fini de tester un script GEE et que vous souhaitez qu'il soit accessible à l'aide du plugiciel QGIS (et par d'autres utilisateurs de Trends.Earth), vous pouvez le déployer sur le serveur api.trends.earth. La première étape du processus consiste à se connecter au serveur api.trends.earth. Pour vous connecter, exécutez :

```
invoke tecli-login
```

Il vous sera demandé un nom d'utilisateur et un mot de passe. Il s'agit du même nom d'utilisateur et du même mot de passe que ceux que vous utilisez pour vous connecter au serveur Trends.Earth à partir du plugiciel QGIS. **Si vous n'êtes pas un administrateur, vous pourrez vous connecter, mais la commande ci-dessous échouera.** Pour télécharger un script (par exemple, le script « land_cover ») sur le serveur, exécutez :

```
invoke tecli-publish -s land_cover
```

Si ce script existe déjà sur le serveur, il vous sera demandé si vous souhaitez écraser le script existant. Soyez très prudent lorsque vous téléchargez des scripts avec des numéros de version pairs, car il s'agit de scripts accessibles au public, et toute erreur que vous commettez affectera toute personne utilisant le plugiciel. Lorsque vous effectuez des tests, veillez à utiliser des numéros de version de développement (numéros de version impairs).

Après avoir publié un script sur le serveur, vous pouvez utiliser la tâche « tecli-info » pour vérifier le statut du script (pour savoir s'il a été déployé avec succès — mais notez que la construction du script peut prendre quelques minutes). Pour vérifier le statut d'un script déployé, exécutez :

```
invoke tecli-publish -s land_cover
```

Si vous faites une nouvelle version du plugiciel, et que vous voulez télécharger TOUS les scripts GEE en une seule fois (ceci est nécessaire lorsque le numéro de version du plugiciel change), exécutez :

```
invoke tecli-publish
```

Encore une fois, n'exécutez jamais la procédure ci-dessus sur une version publique du plugin, sauf si vous avez l'intention d'écraser tous les scripts publics utilisés par le plugin.

1.3 Modification des modèles de couches vectorielles

Trends.Earth permet aux utilisateurs de numériser de nouveaux éléments vectoriels pour délimiter des zones d'intérêt particulier.

Pour l'instant, seules les couches « faux positifs/négatifs » sont supportées, mais d'autres peuvent être ajoutées si nécessaire. Toute couche vectorielle est créée à partir des fichiers GeoPackage modèles, qui se trouvent dans le dossier « data/error_recode » du répertoire d'installation du plugiciel. Pour chaque type de vecteur, il y a 6 fichiers modèles, un pour chaque langue officielle de l'ONU. Le code de langue ISO est ajouté comme suffixe au nom du fichier. Ceci est nécessaire pour fournir des étiquettes localisées dans les formulaires d'attributs. Lorsque la création d'une couche vectorielle est demandée, QGIS recherche le fichier modèle en tenant compte des paramètres régionaux de QGIS ; comme option de repli, la version anglaise du fichier modèle est utilisée.

Pour changer le schéma de la couche, il est nécessaire de modifier les fichiers modèles correspondants dans le dossier « data/error_recode » du répertoire d'installation du plugiciel. Le fichier modèle contient également un style par défaut intégré et une configuration du formulaire d'attribut qui sera automatiquement appliqué à la couche lors du chargement dans QGIS.

Pour afficher les graphiques sous forme d'attributs, un gadget logiciel QML intégré est utilisé. Les données des graphiques sont stockées dans la table d'attributs de la couche vectorielle. Les valeurs des champs correspondants sont extraites à l'aide d'expressions.

Le code pour générer les graphiques ressemble à ceci :

```
import QtQuick 2.0
import QtCharts 2.0

ChartView {
    width: 380
    height:200
    margins {top: 0; bottom: 0; left: 0; right: 0}
    backgroundColor: "#eeeeec"
    legend.alignment: Qt.AlignBottom
    antialiasing: true
    ValueAxis {
        id: valueAxisY
        min: 0
        max: 100
    }

    BarSeries {
        id: mySeries
        axisY: valueAxisY
        axisX: BarCategoryAxis { categories: ["Productivity", "Land cover", "Soil_
↵organic carbon"] }
        BarSet { label: "Degraded"; color: "#9b2779"; values: [expression.evaluate("\
↵prod_deg\""), expression.evaluate("\land_deg\""), expression.evaluate("\soil_deg\
↵")] }
        BarSet { label: "Improved"; color: "#006500"; values: [expression.evaluate("\
↵prod_imp\""), expression.evaluate("\land_imp\""), expression.evaluate("\soil_imp\
↵")] }
        BarSet { label: "Stable"; color: "#ffffe0"; values: [expression.evaluate("\
↵prod_stab\""), expression.evaluate("\land_stab\""), expression.evaluate("\soil_
↵stab\"")] }
    }
}
```

Pour extraire la valeur du champ, la fonction « expression.evaluate(»"prod_deg" ») » est utilisée, le seul argument qu'elle accepte est le nom du champ. Pour les couches faussement positives/négatives, le graphique contient trois indicateurs : productivité, couverture du sol et carbone organique du sol. Pour chaque indicateur, le plugiciel conserve trois valeurs : le pourcentage stable, dégradé et amélioré de la surface du polygone. Par exemple, dans le cas de l'indicateur de productivité, les champs seront :

- prod_deg - productivité dégradée
- prod_stab - productivité stable
- prod_imp - amélioration de la productivité

La même méthode de dénomination est appliquée à la couverture terrestre (champs « land_* ») et au carbone organique du sol (champs « soil_* »).

Le calcul du pourcentage de surface est fait avec une fonction d'expression personnalisée, son code se trouve dans le fichier

« charts.py » dans le répertoire racine du plugiciel. La fonction est optimisée pour travailler avec de grands polygones et utilise le flux de travail suivant. Pour une géométrie donnée, trouver une bbox et extraire le sous-ensemble raster en utilisant cette bbox. Effectuer le rastering de la géométrie en mémoire et l'appliquer comme un masque au raster. Ensuite, compter le nombre de pixels qui ont une valeur spécifique et calculer le pourcentage. Comme le comptage des pixels est construit sur des fonctions de tableau numpy, il est très rapide même pour les grands polygones.

Lors de la première tentative d'édition d'une couche vectorielle, l'utilisateur se verra présenter une boîte de dialogue dans laquelle il devra sélectionner les jeux de données à utiliser pour les indicateurs. Ensuite, le plugiciel configurera des valeurs d'expression par défaut pour tous les champs d'indicateurs, de sorte que la valeur sera mise à jour à chaque changement de géométrie.

1.4 Traitement des métadonnées des ensembles de données

Les métadonnées du jeu de données sont stockées dans le format QMD de QGIS. Ces fichiers QMD peuvent être créés pour chaque raster individuellement et aussi pour l'ensemble du jeu de données. Le dialogue de l'éditeur de métadonnées s'ouvre à partir du menu **Editer les métadonnées** dans le dock de Trends.Earth.

Lorsque le jeu de données est exporté au format ZIP, la conversion en ISO XML est effectuée à l'aide d'une transformation XSLT. Les transformations correspondantes sont situées dans le sous-répertoire « dataxsl » du dossier d'installation du plugin.

1.5 Mise à jour du cadre de présentation des rapports

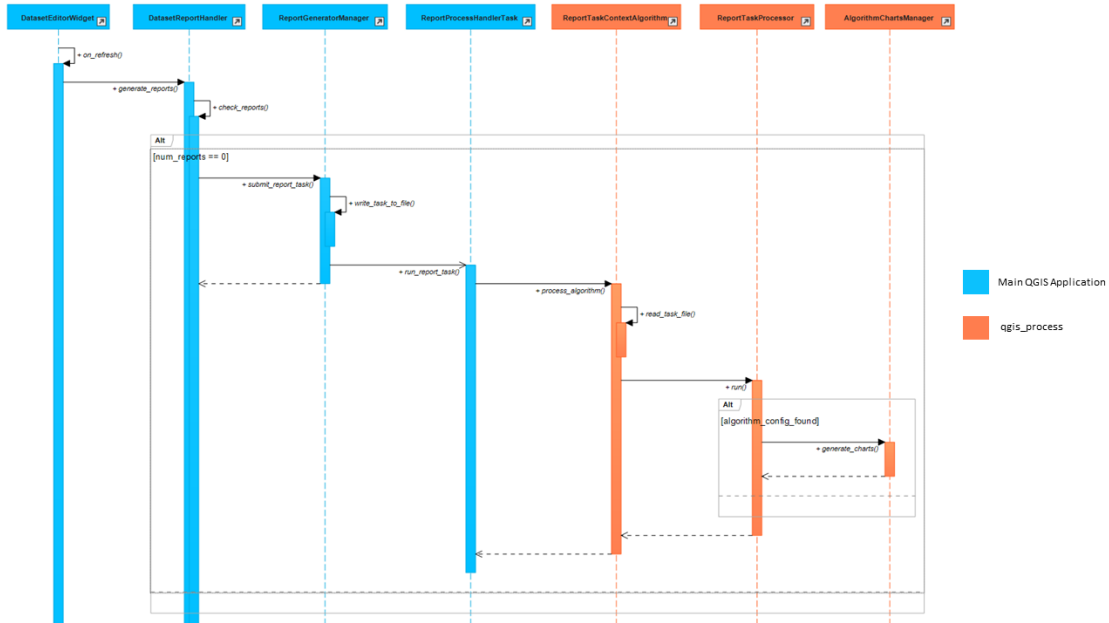
1.5.1 Aperçu du cadre de présentation des rapports

Le cadre de rapports est conçu pour être extensible tout en offrant une interactivité à l'utilisateur grâce à des opérations non bloquantes. Le Framework s'appuie fortement sur les classes `QgsProject` et `QgsPrintLayout` qui ne sont pas thread safe d'où l'utilisation de `qgis_process` pour faire le gros du travail de génération des rapports (et des graphiques). Vous pouvez trouver plus d'informations sur `qgis_process` [ici](#).

La boîte à outils effectue deux étapes principales lors de la génération de rapports (et de graphiques) pour les couches par défaut d'un travail :

1. Il crée un objet `ReportTaskContext` qui constitue un objet `ReportConfiguration` (voir `config_report_params`) et un objet `Job` qui est représenté dans le panneau **Datasets**. Cet objet `ReportTaskContext` est sérialisé dans un fichier JSON puis transmis comme l'un des arguments d'un objet `ReportProcessHandlerTask` (qui hérite de `QgsTask`).
2. L'objet `ReportProcessHandlerTask` initie une instance distincte de `qgis_process` et transmet le chemin d'accès au fichier JSON comme entrée à l'algorithme de traitement `trendsearch:reporttask`. Il s'agit d'une enveloppe fine qui déserialise le fichier vers l'objet `ReportTaskContext` et le transmet à un objet `ReportTaskProcessor` chargé de générer les rapports et le projet QGIS de la tâche. Pour les algorithmes qui nécessitent des graphiques, l'objet `ReportTaskProcessor` transmet l'objet du travail à un objet `AlgorithmChartsManager` qui vérifie si une configuration de graphique est définie pour l'algorithme du travail. Si elle est définie, il génère les graphiques correspondants sous forme de fichiers PNG. (Voir [Ajout de configurations de graphiques](#) pour plus d'informations sur les configurations de graphiques)

Le diagramme ci-dessous fournit une illustration de haut niveau de ce processus :



Cliquez sur l'image pour l'agrandir.

Note: Certains des noms de fonctions dans le diagramme ci-dessus ont été simplifiés à des fins d'illustration. Les classes mentionnées ci-dessus se trouvent dans les modules `LDMP.reports` et `LDMP.processing_provider.report`.

1.5.2 Ajout de variables de mise en page du rapport

Les variables de rapport fournissent des informations contextuelles liées à un travail, une couche (ou une bande) ou un `report_settings` pendant le processus d'exécution du rapport. Actuellement, la boîte à outils prend en charge les variables répertoriées dans la section `layout_expr_vars`.

Chaque variable est définie comme un `namedtuple` dans le module `LDMP.reports.expressions` et est ensuite mise à jour et évaluée par l'objet `ReportTaskProcessor`.

Suivez les directives ci-dessous sur la façon d'ajouter de nouvelles variables de travail ou de couche actuelle.

Poste variable

Il permet d'ajouter des informations sur le travail en cours d'exécution à la mise en page d'un rapport. Les informations relatives à chaque variable de travail sont encapsulées dans un objet `JobAttrVarInfo` composé de quatre attributs :

Nom de l'attribut	Description	Type de données	Valeur par défaut
<code>job_attr</code>	Nom de l'attribut d'un objet <code>Job</code> tel qu'utilisé dans une notation par points. Par exemple, <code>id</code> correspond à <code>job.id</code> . Vous pouvez même utiliser la notation par points pour faire référence aux attributs des classes internes imbriquées, par exemple <code>results.uri.uri</code> .	Chaîne de caractères	N/A
<code>var_name</code>	Nom de la variable de mise en page du rapport. Il doit être précédé du préfixe <code>te_job_</code> .	Chaîne de caractères	N/A
<code>default_val</code>	Une valeur par défaut à utiliser pour <code>var_name</code> , principalement appliquée lors de la conception des layouts.	sujet	sujet
<code>fmt_func</code>	Un objet fonction qui sera utilisé pour convertir la valeur de l'attribut du travail dans un format compatible avec les expressions QGIS. Par exemple, <code>str</code> peut être utilisé pour convertir la valeur de l'attribut <code>id</code> d'un travail de UUID en chaîne de caractères. Vous pouvez également utiliser des fonctions lambda ici.	objet de la fonction	Néant

L'extrait de code ci-dessous montre comment ajouter une variable `te_job_result_name` qui correspond à `job.results.name`.

```
# LDMP/reports/expressions.py
def _job_attr_var_mapping() -> typing.List[JobAttrVarInfo]:
    return [
        ...
        JobAttrVarInfo('results.name', 'te_job_result_name', '', str),
        ...
    ]
```

Variable de la couche

Il fournit des informations sur la couche matricielle en cours d'exécution. Ces informations variables sont encapsulées dans un objet `LayerVarInfo` composé de trois attributs :

Nom de l'attribut	Description	Type de données	Valeur par défaut
<code>var_name</code>	Nom de la variable de mise en page du rapport. Il doit être précédé du préfixe <code>te_current_layer_</code> .	Chaîne de caractères	N/A
<code>default_val</code>	Une valeur par défaut à utiliser pour <code>var_name</code> , principalement appliquée lors de la conception des layouts.	sujet	sujet
<code>fmt_func</code>	Un objet fonction qui sera utilisé pour extraire et/ou convertir une valeur d'un objet <code>QgsRasterLayer</code> dans un format compatible avec les expressions QGIS. Vous pouvez également utiliser des fonctions lambda ici. Par exemple, <code>lambda layer : layer.name()</code> renvoie le nom de la couche.	objet de la fonction	Néant

L'extrait de code ci-dessous montre comment ajouter une variable `te_current_layer_height` qui correspond à la hauteur de la couche matricielle.

```
# LDMP/reports/expressions.py
def _current_job_layer_var_mapping() -> typing.List[LayerVarInfo]:
    return [
        ...
        LayerVarInfo(
            'te_current_layer_height',
            '',
            lambda layer: layer.height()
        )
        ...
    ]
```

Note: Ces variables ne sont disponibles que dans la portée de la mise en page.

1.5.3 Ajout de configurations de graphiques

Les graphiques peuvent être regroupés à l'aide d'un objet de configuration de graphique qui correspond à un algorithme spécifique. La définition d'une nouvelle configuration de graphique est un processus en trois étapes :

1. Créez une nouvelle classe de graphique qui hérite de `BaseChart` dans le module `LDMP.reports.charts`. Implémentez la fonction `export` pour spécifier le type de graphique, les propriétés, etc. en utilisant la bibliothèque Python `Plotly` fournie avec QGIS. Enfin, dans la fonction `export`, appelez la fonction `save_image` pour écrire l'objet `Figure` de `Plotly` sous forme de fichier image en utilisant l'un des formats pris en charge par la classe `QImageWriter` de Qt. Vous pouvez également spécifier le chemin relatif au répertoire de sortie racine, qui est également disponible comme attribut dans la classe de base. Voir l'extrait de code ci-dessous :

```
# LDMP/reports/charts.py
class MyCustomChart(BaseChart):
    def export(self) -> typing.Tuple[bool, list]:
        status = True
        messages = []

        # Create chart Figure using Plotly and set properties
        fig = go.Figure(...)

        # Add warning or error messages
        messages.append('Colour list not supported.')

        # Set image path in dataset's reports folder
        img_path = f'{self.root_output_dir}/chart-NDVI.png'

        # Save image and append its path
        self.save_image(fig, img_path)
        self._paths.append(img_path)

        return status, messages
```

Vous pouvez vous inspirer de la classe `UniqueValuesPieChart` pour un exemple plus complet.

2. Créez une classe de configuration de graphiques qui hérite de `BaseAlgorithmChartsConfiguration` et implémentez la fonction `_add_charts`. La classe de configuration des graphiques définit essentiellement les graphiques qui seront utilisés pour un algorithme donné. L'attribut `layer_band_infos` est une liste d'objets `LayerBandInfo` qui contient les données relatives à la couche et à la bande nécessaires à la production des

graphiques. Vous pouvez vous inspirer de la classe `LandCoverChartsConfiguration` pour un exemple plus complet.

- Enfin, faites correspondre un algorithme (nom) à la classe de configuration de graphique correspondante dans la classe `AlgorithmChartsManager` comme indiqué ci-dessous :

```
# LDMP/reports/charts.py
class AlgorithmChartsManager:
    def _set_default_chart_config_types(self):
        ...
        self.add_alg_chart_config('land-cover', LandCoverChartsConfiguration)
        self.add_alg_chart_config('productivity',
        ↪MyCustomLandProductivityChartsConfiguration)
        ...
```

La classe `AlgorithmChartsManager`, instanciée dans l'objet `ReportTaskProcessor`, crée un nouvel objet de configuration de graphique pour l'algorithme d'une tâche correspondante lorsque des rapports sont générés.

1.6 Contribuer à la documentation

1.6.1 Aperçu général

La documentation de Trends.Earth est produite à l'aide de `Sphinx`, et est écrite au format `reStructuredText`. Si vous n'êtes pas familier avec l'un ou l'autre de ces outils, consultez leur documentation pour plus d'informations sur leur utilisation.

La documentation de Trends.Earth est stockée dans le dossier « docs » sous le répertoire principal de trends.earth. Dans ce dossier, il y a un certain nombre de fichiers et dossiers clés à connaître :

- `build` : contient la documentation de compilation pour trends.earth (au format PDF et HTML). Notez qu'elle n'apparaîtra sur votre machine qu'après avoir exécuté la tâche `invoke docs-build`.
- `i18n` : contient les traductions de la documentation dans d'autres langues. Les fichiers contenus dans ce dossier sont normalement traités automatiquement à l'aide de tâches d'invocation, de sorte que vous ne devriez jamais avoir de raison de modifier quoi que ce soit dans ce dossier.
- `ressources` : contient toutes les ressources (principalement des images ou des PDF) auxquelles il est fait référence dans la documentation. Actuellement, il n'y a qu'un seul dossier (« EN », pour anglais) car toutes les images de la documentation proviennent de la version anglaise du logiciel — si nécessaire, des dossiers supplémentaires peuvent être ajoutés sous « ressources » avec des codes de langue à deux lettres pour inclure les images spécifiques à une langue particulière.
- `source` : contient les fichiers source de `reStructuredText` qui définissent la documentation (il s'agit du texte anglais réel de la documentation, et ce sont les fichiers que vous devrez probablement modifier).

1.6.2 Installation des dépendances

Dépendances Python

Afin de travailler avec la documentation, vous devez avoir `invoke`, `Sphinx`, `sphinx-intl`, et `sphinx-rtd-theme` (le thème du site web Trends.Earth) installés sur votre machine. Ces paquets sont tous listés dans le fichier d'exigences « dev » pour Trends.Earth, ils peuvent donc être installés en naviguant dans une invite de commande à la racine du dossier de code de trends.earth et en tapant :

```
pip install -r requirements-dev.txt
```

LaTeX

LaTeX est utilisé pour produire des sorties PDF de la documentation de Trends.Earth.

Pour installer sur Windows, suivez le processus décrit ici pour installer la distribution proTeXt de LaTeX à partir du fichier zip disponible ici. Le programme d'installation de LaTeX est assez volumineux (plusieurs Go) et le téléchargement et l'installation peuvent prendre un certain temps.

Sous MacOS, MacTeX est une bonne option, et peut être installé en suivant les instructions ici <<http://www.tug.org/mactex/>>`_.

Sous Linux, l'installation de LaTeX devrait être beaucoup plus facile - utilisez le gestionnaire de paquets de votre distribution pour trouver et installer la distribution LaTeX incluse par défaut.

Qt Linguist

Qt Linguist est également nécessaire afin d'extraire les chaînes du code et de l'interface graphique pour les traduire. La commande « lrelease » doit être disponible et sur votre chemin. Essayez d'essayer :

```
lrelease
```

dans une fenêtre de terminal. Si le fichier n'est pas trouvé, vous devrez installer Qt Linguist. Cette page <<https://github.com/lelegard/qtlinguist-installers/releases>>`_ est une source d'installateurs pour Qt Linguist. Une fois que vous avez installé Qt Linguist, assurez-vous d'ajouter le dossier contenant lrelease à votre chemin afin que le script invoque de Trends.Earth puisse le trouver.

1.6.3 Mise à jour et élaboration de la documentation

Une fois que vous avez installé la configuration requise pour sphinx, vous êtes prêt à commencer à modifier la documentation. Les fichiers à modifier sont situés dans le dossier « docs\source ». Après avoir apporté des modifications à ces fichiers, vous devrez construire la documentation afin de visualiser les résultats. Il existe deux versions de la documentation de Trends.Earth : une version HTML (utilisée pour le site web) et une version PDF (pour le téléchargement hors ligne). Pour construire la documentation de Trends.Earth, utilisez la tâche invoke « docs-build ». Par défaut, cette tâche va construire la documentation complète de Trends.Earth, en HTML et PDF, pour toutes les langues prises en charge. L'exécution de cette tâche peut prendre un certain temps (jusqu'à quelques heures). Si vous voulez juste tester les résultats de quelques changements mineurs à la documentation, il est généralement préférable d'utiliser l'option « f » (pour « fast »). Cette option ne construira que la documentation HTML en anglais, ce qui ne devrait prendre que quelques secondes. Pour construire en utilisant l'option rapide, exécutez :

```
invoke docs-build -f
```

La commande ci-dessus prendra quelques secondes à s'exécuter, puis si vous regardez sous « docs\buildhtml\en », vous verrez la version HTML de la documentation. Chargez le fichier « index.html » dans un navigateur Web pour voir à quoi il ressemble.

Pour construire la documentation complète, pour toutes les langues, en PDF et en HTML (rappelez-vous que cela peut prendre quelques heures), exécutez :

```
invoke docs-build
```

Après avoir exécuté la commande ci-dessus, vous verrez (pour l'anglais) la documentation HTML sous « docs\build\html\en », et les PDFs de la documentation sous « docs\build\html\en\pdfs ».

Si vous voulez tester une langue spécifique (lorsque vous testez des traductions, par exemple), vous pouvez spécifier un code de langue à deux lettres pour ne construire que les documents de cette langue. Par exemple, pour construire uniquement la documentation en espagnol, exécutez :

```
invoke docs-build -l es
```

Notez que les options peuvent être combinées, ainsi vous pouvez utiliser l'option rapide pour construire uniquement la version HTML de la documentation en espagnol en exécutant :

```
invoke docs-build -f -l es
```

Lors de la construction de la documentation complète du site web, il est bon de supprimer d'abord toutes les anciennes constructions de la documentation, car elles peuvent contenir des fichiers qui ne sont plus utilisés dans la documentation mise à jour. Pour ce faire, utilisez l'option « -c » (clean) :

```
invoke docs-build -c
```

En général, docs-build DOIT se terminer sans aucune erreur si vous envisagez de partager la documentation ou de la publier sur le site Web. Cependant, lorsque vous testez les choses localement, vous pouvez vouloir ignorer les erreurs de documentation qui apparaissent seulement pour certaines langues (dus à des erreurs de syntaxe résultant d'erreurs de traduction, etc.), et continuer à construire le reste de la documentation sans tenir compte des erreurs. Pour ce faire, utilisez l'option « -i » (ignorer les erreurs) :

```
invoke docs-build -i
```

Chaque fois que vous apportez des modifications au texte de la documentation, il est bon de pousser les dernières chaînes de caractères vers Transifex afin qu'elles puissent être traduites. Pour mettre à jour les chaînes de caractères sur Transifex avec les nouveaux changements, exécutez :

```
invoke translate-push
```

Note: Pour exécuter la commande ci-dessus, vous devez disposer de la clé du compte transifex de Trends.Earth.

1.6.4 Construction de la documentation pour la version

Avant de publier une nouvelle documentation, tirez toujours les dernières traductions de Transifex afin que toutes les traductions soient à jour. Pour ce faire, exécutez :

```
invoke translate-pull
```

Pour construire une version de la documentation pour une diffusion publique (soit sur le site web, soit en PDF), vous devez construire la documentation entière en utilisant docs-build sans paramètres supplémentaires :

```
invoke docs-build
```

Ce processus doit se terminer avec succès et sans erreur. Si des erreurs surviennent au cours du processus, examinez le message d'erreur et apportez les modifications nécessaires pour permettre à la compilation de se terminer avec succès. Une fois la compilation terminée sans erreur, les fichiers sont prêts à être déployés sur le site Web.

Note: Les deux commandes ci-dessus ont également des options « -f » (force) qui forcent à tirer ou pousser les dernières traductions depuis ou vers Transifex (respectivement). N'utilisez ces options que si vous êtes TRÈS sûr de ce que vous faites, car elles peuvent écraser complètement les traductions sur Transifex, entraînant une perte du travail effectué par les traducteurs si les dernières traductions n'ont pas encore été livrées sur github.

1.6.5 Ajout d'un nouveau texte de documentation

Tous les nouveaux fichiers .rst ajoutés à la documentation doivent être ajoutés à plusieurs fichiers de configuration pour s'assurer qu'ils apparaissent dans le menu de navigation, qu'ils sont correctement traduits et (pour les didacticiels) pour s'assurer qu'ils sont générés en PDF afin qu'ils puissent être téléchargés pour une utilisation hors ligne.

- docs\sourceindex.rst : ajoutez de nouveaux fichiers .rst à l'endroit approprié ici pour vous assurer qu'ils sont liés au menu de navigation.
- .tx\config : listez ici les nouveaux fichiers .rst (dans le même format que les autres fichiers déjà inclus) afin de les faire connaître au logiciel de traduction pour qu'ils puissent être traduits.
- docs\sourceconf.py : si vous voulez générer un fichier PDF d'une page du site web, alors vous devez lister cette page ici dans la liste « latex_documents ». Habituellement, nous ne faisons cela que pour les pages de tutoriel que nous voulons mettre à la disposition des participants à l'atelier dans des PDF individuels. Chaque page du site sera incluse dans la version PDF du site dans son ensemble, qu'elle figure ou non dans la liste « latex_documents ».

1.6.6 Ajout de nouvelles images ou d'autres ressources

Toute nouvelle image ou autre ressource (PDF, etc.) nécessaire à la documentation doit être ajoutée dans la rubrique « docs_resources_en ». Si vous le souhaitez, il est possible de télécharger différentes versions d'une image afin que celle-ci apparaisse avec les traductions appropriées. Cela peut être utile si vous souhaitez afficher l'interface graphique dans la langue appropriée, par exemple. Pour ce faire, téléchargez d'abord une copie de l'image dans « docs\\resourcesen » (avec du texte en anglais). Ensuite, créez une copie de l'image avec le texte traduit et placez cette image dans le dossier approprié pour cette langue (par exemple, une image présentant des traductions en espagnol serait placée dans « docs/resources\es »). La version anglaise de l'image sera utilisée par défaut pour toutes les langues pour lesquelles une version native de l'image n'est pas fournie, tandis qu'une version localisée sera utilisée lorsqu'elle sera disponible.

Note: Il existe un autre dossier, « docs\source\static », qui est utilisé pour conserver temporairement les ressources pendant l'exécution des scripts qui construisent la documentation Trends.Earth. Vous pouvez avoir des images listées dans ce dossier si vous avez déjà construit la documentation sur cette machine. **Ce dossier ne doit jamais être utilisé pour ajouter de nouvelles ressources** — les nouvelles ressources doivent toujours être placées dans le dossier « docs/resources\en » ou, pour les images traduites, dans le dossier spécifique à la langue appropriée dans le dossier « docsresources ».

1.6.7 Contribuer en tant que traducteur

Les traductions du logiciel QGIS et de ce site sont gérées par [transifex](#). Si vous souhaitez contribuer à la traduction du logiciel et de la documentation (et nous serions ravis d'avoir votre aide !), vous pouvez demander à rejoindre [notre équipe](#) par le biais de [transifex](#), ou en nous envoyant un courriel à trends.earth@conservation.org.

Journal des modifications

Cette page présente l'historique des versions deltrends.earthl.

2.1 2.1.8 (January 25, 2023)

- Output JSON field for Prais containing updated false positive/negative calculations.
- Fixes #752 - false positive/negative datasets not showing up in non-English versions

2.2 2.1.6 (December 8, 2022)

- Bumps maximum number of land cover classes to 45 (useful particularly for users seeking to use Corine legend)
- Fix issue that was arising on Mac/Linux due to missing QgsPanelWidget ui file
- Various more minor bug fixes

2.3 2.1.6 (November 21, 2022)

- Fixes #691 (issue that arose if the base data directory was cleared). When base directory is cleared plugin will now reset the value to the default.
- Fix python error coming up when settings window was closed while datasets screen was not open.

2.4 2.1.2 (November 18, 2022)

- Fixes #724 - ensure max number of land cover classes is set to 38 by default (ignoring the previously set default value in qsettings, which used to be 32).
- Fixes #722 - file name casing when code refers to land_cover_transition_matrix_unccd.json.
- Fixes #726 - make dataset year visible in datasets window for imported datasets

2.5 2.1.0 (November 11, 2022)

- Moved plugin settings to live within main QGIS plugin settings window
- Bumped marshmallow-dataclass to 8.5.10 and marshmallow to 3.18.0. This necessitated changes in trends.earth-schemas, so it is possible some old jobs may not load in this new version (suggest re-running them if need be).

2.6 2.0.7 (October 31, 2022)

- Enable Excel table generation for SDG 15.3.1 when using custom classes - all the various minor bugs should now be worked out.
- Limit report generation to 3 tries - then quit trying to generate reports until next restart (prevents Trends.Earth continually spawning processes for report generation when they are failing).

2.7 2.0.5 (October 19, 2022)

- Fix SDG summary generation when using JRC LPD or WOCAT LPD.

2.8 2.0.3 (October 19, 2022)

- Allow bypassing reset of land cover legend when choosing defaults in the SDG tool.
- Disable excel generation when child legend does not have 7 classes.
- Fix report generation when using custom legend.

2.9 2.0.1 (October 13, 2022)

- Added ability to completely customize the land cover legend used in Trends.Earth, including the number, name, color, and coding of each class in the legend. This applies to all calculatons using land cover data, including the SDG 15.3.1 tool, and the productivity, soil organic carbon, and land cover indicator tools.
- Reworked land cover import tool to allow importing data and assigning classes to custom legend.
- Misc minor bug fixes.
- Add ability to include false positive/negative dataset in UNCCD report. Next release will include further functionality to apply and report on false positive/negatives.

- This experimental release disables the creation of Excel output tables for the SDG 15.3.1 tool. This functionality will be present in the next (stable) release.

2.10 2.0 (July 20, 2022)

- Une toute nouvelle interface avec des écrans de navigation et un chargement des données simplifiés
- Documentation et site Internet mis à jour
- Trends.Earth est désormais traduit dans toutes les langues officielles de l'ONU, ainsi qu'en portugais. Des traductions en swahili et en farsi (pour la nouvelle interface) sont également prévues.
- Nouvelles fonctions pour soutenir le dernier cycle de rapports de la CNULCD (y compris le soutien direct pour l'exportation des résultats pour le téléchargement vers le système PRAIS de la CNULCD)
- Nouvelles fonctions pour soutenir l'évaluation des risques, de la vulnérabilité et de l'exposition à la sécheresse
- Nouvelles fonctions de l'outil de séries chronologiques (supporte maintenant le traçage des tendances résiduelles, l'EUE, etc.)

2.11 1.0.10 (7 juillet 2022)

- Correction d'un bug avec TransformDirection lié aux changements dans QGIS 3.22+

2.12 1.0.8 (15 octobre 2021)

- Correction d'un bug dans la sauvegarde de la couche finale pour le calcul des ODD (lié aux problèmes n° 500 et n° 505)

2.13 1.0.6 (15 juillet 2021)

- Suppression du sous-module trends.earth-schemas et installation par le biais de setup.py (beaucoup plus efficace pour le suivi des versions, le développement, etc.)
- Correction d'un bug dans le chargement des données NDVI 2020 dans le code GEE (le nom de l'actif n'était pas bien défini et l'année 2020 ne se chargeait pas)
- Ajout de diverses tâches d'invocation pour faciliter le développement et la sortie des plugins.

2.14 1.0.4 (30 juin 2021)

- Ajout de WorldPop et Gridded Population of the World version 4 (GPWv4) aux ensembles de données disponibles sur Trends.Earth
- Mise à jour pour permettre l'accès à la couverture terrestre ESA-CCI jusqu'en 2020
- Mise à jour des ensembles de données GPCC et GPCP
- Correction d'un bug quand des tampons non entiers sont utilisés
- Correction de coquilles dans la documentation

2.15 1.0.2 (14 août 2020)

- Correction du code de zone urbaine pour permettre le traitement des sites d'intérêt dont la superficie est comprise entre 10 000 km² et 25 000 km².
- Ajout des dernières données MERRA2 (jusqu'en 2019).
- Suppression de la limitation de la zone maximale de l'outil de téléchargement.
- Passage de httplib2 à 0.18.0.
- Passage de GPCC V6 à GPCC V7
- Ajout des données Hansen et al. deforestation de 2019
- Passage à la dernière carte de couleurs pour les données de dégradation (résolution des problèmes de daltonisme rouge/vert).
- Prise en charge de plus de données dans les fichiers de formes d'entrée (ajout de PointZ, MultiPoint, MultiPointZ, PolygonZ, MultiPolygonZ).
- Diverses corrections de bugs pour résoudre les erreurs Python qui accompagnaient certains messages QMessageBox.

2.16 1.0.0 (27 avril 2020)

- Possibilité de télécharger et d'utiliser des binaires pré-compilés (compilés avec Numba) pour accélérer certains calculs locaux. Ce n'est actuellement disponible que pour le calcul du tableau récapitulatif SDG15.3.1, mais cela sera à terme étendu à d'autres outils.
- En outre, il existe désormais un bouton de paramètres « avancés » dans la fenêtre des paramètres. Il permet aux utilisateurs de télécharger des binaires pré-compilés, et d'activer ou de désactiver l'enregistrement de messages lorsque l'outil fonctionne. Ces messages peuvent aider à résoudre des problèmes le cas échéant.
- Amélioration des contrôles de validité des géométries dans les fichiers de saisie, et affichage d'un message d'erreur plutôt que de lancer une exception en cas d'erreurs de géométrie.
- Correction du traitement utilisant les nouvelles données MODIS (les actifs n'étaient pas correctement mis à jour dans la version précédente)
- Correction du codage couleur des couvertures terrestres dans la fenêtre d'agrégation des classes de couverture terrestre
- Ajout à la boîte de dialogue « à propos » d'informations plus détaillées sur la version.
- Ajout de détails supplémentaires à l'outil de téléchargement des données.
- Ajout de la référence du travail dans la fenêtre de téléchargement pour faciliter le signalement de bugs.
- Corrections de bugs mineurs (tri des tableaux de travaux et de téléchargements).

2.17 0.98 (2 avril 2020)

- Première version de QGIS3 ; nombreuses corrections pour passer à Qt5 et à l'API QGIS3.
- Passage de toutes les dépendances du plugin aux dernières versions à partir de janvier 2020.
- Correction de l'outil de téléchargement de données pour avoir 1style par défaut pour tous les ensembles de données disponibles.
- Début de la migration vers les cadres QgsProcessing et QgsTask ; actuellement, seul l'outil carbone a été migré, mais tous les outils le seront avant le passage à la version 1.0./
- Nettoyage du formatage de la feuille de calcul pour l'outil carbone afin de faciliter la compréhension de chaque colonne.
- Mise à jour de tous les scripts GEE pour utiliser la dernière version de l'API GEE (0.1.213).
- Sauvegarde de plus de paramètres choisis dans les boîtes de dialogue des outils entre les sessions QGIS.
- Nettoyage du code de mise en mémoire tampon, pour utiliser les projections de Lambert Equal Area centrées sur les centroïdes des polygones pour la mise en mémoire tampon.
- Déplacement de la documentation dans le dossier docs à la racine du référentiel trends.earth.
- La documentation comporte davantage de détails sur la façon de contribuer au développement de Trends.Earth.
- Nettoyage du référentiel par la suppression des fichiers de traduction compilés et l'ajout de ces types de fichiers à .gitignore.
- Le projet transifex est renommé « trendsearth ».
- Diverses corrections d'erreurs de compatibilité et de bugs mineurs.

2.18 0.66 (20 juillet 2019)

- Limitation de la surface maximale des tâches à 10 000 000 km², sauf pour les tâches en zone urbaine, qui sont limitées à 10 000 km².
- Ajout d'une section sur SDG 11.3.1 et mise à jour du tutoriel SDG 11.3.1.
- Mise à jour du code SDG 11.3.1 pour inclure l'année 1998 dans la série (en interne lors du calcul) afin de filtrer le bruit au début de la série urbaine.
- Correction des restrictions de dates dans l'outil tout-en-un SDG 15.3.1 pour tenir compte de la disponibilité de l'ESA et de MODIS.
- Ajout d'une section au readme sur la façon d'installer les nouveautés Github.
- Mise à jour et révision des traductions espagnoles ; mise à jour des traductions google pour les autres langues.

2.19 0.64 (9 juillet 2019)

- Correction de la manipulation des nodata dans l'outil de carbone total.
- Prise en charge des données Hansen de 2018 dans l'outil de carbone total.
- Prise en charge des données mondiales sur la biomasse 30m provenant de Wood's Hole dans le carbone total
- 2015 est définie comme l'année finale maximale pour l'outil SDG 15.3.1 une étape (ce qui correspond aux données de l'ESA).
- La productivité Trends.Earth est devenue l'ensemble de données par défaut dans l'outil SDG une étape pour 15.3.1.

2.20 0.62 (27 janvier 2019)

- Ajout d'un outil expérimental pour cartographier les rendements en carbone potentiels à partir des interventions de restauration alternatives.
- Ajout des données MODIS 2018.
- Diverses corrections de la dimension des fenêtres dans l'interface graphique.
- Passage à la dernière version d'openpyxl ; corrections du chargement du logo Trends.Earth dans les tableaux récapitulatifs.
- Ajout de la liste des publications aux documents d'aide.

2.21 0.60 (3 décembre 2018)

- Le calcul du changement de la zone urbaine et du taux de croissance de la population a été ajouté (SDG 11.3.1)
- Correction des hauteurs par défaut des boutons/champs de saisie Ajout de la sélection de la ville pour les sites d'intérêt
- Ajout d'une mise en tampon optionnelle pour les sites d'intérêt

2.22 0.58 (11 août 2018)

- Ajout d'une section de test à la page des calculs
- Ajout d'une version d'essai du carbone total (au-dessus et au-dessous du sol) et des émissions dues à la déforestation
- Correction de bugs mineurs, y compris pour les polygones invalides dans les sites d'intérêt d'entrée

2.23 0.56.5 (21 mai 2018)

- Correction d'une erreur avec l'importation LPD demandant une année de données.

2.24 0.56.4 (21 mai 2018)

- Toujours rééchantillonner les données importées à la plus haute résolution.
- Correction de l'importation personnalisée de zones climatiques par SOC afin d'utiliser un ensemble de données de zones climatiques élargi pour éliminer l'absence de données.
- MOD16A2 est mis à jour avec les dernières données.
- Saisie forcée de la date lors de l'importation de données SOC et LC
- Ajout des sorties mondiales de Trends.Earth à l'outil de téléchargement.
- Correction de la gestion des valeurs NULL dans les légendes.

2.25 0.56.3 (21 avril 2018)

- Correction du calcul des tableaux récapitulatifs pour les sites d'intérêt répartis sur le 180e méridien (Fidji, Russie, etc.).
- Modification du calcul de l'état afin que les zones présentant des changements de très faible magnitude dans l'intégrale NDVI ($< 0,01$ unité NDVI sur toute la période) soient considérées comme stables.

2.26 0.56.2 (10 avril 2018)

- Corrections mineures de l'unicode.

2.27 0.56.1 (10 avril 2018)

- Correction de l'erreur marshmallow sur le chargement du plugin

2.28 0.56 (9 avril 2018)

- Correction d'un problème de rasterisation des données (trames vides en sortie)
- Forcer l'utilisateur à choisir la résolution de sortie s'il rasterise un vecteur
- Prise en charge du calcul de la dégradation du SOC à partir de données SOC et LC personnalisées

2.29 0.54 (8 avril 2018)

- Prise en charge du chargement de données LPD, SOC et LC personnalisées.
- Nettoyage des styles pour qu'ils correspondent à maps.trends.earth
- Mise à jour de pyopenxl
- Ajout d'icônes d'importation/chargement à toutes les boîtes de sélection de couches.

2.30 0.52.1 (21 mars 2018)

- Corrections de bogues mineurs pendant l'atelier d'Antalya.

2.31 0.52.1 (21 mars 2018)

- Corrections de bogues mineurs pendant l'atelier d'Antalya.

2.32 0.52 (19 mars 2018)

- Code de traitement AOI propre.

2.33 0.50 (15 mars 2018)

- Passer l'exception si seulement liée à l'ajout du logo Trends.Earth dans le fichier Excel.
- Diverses corrections de bogues mineurs.

2.34 0.48 (13 mars 2018)

- Correction du formatage des tableaux

2.35 0.46 (13 mars 2018)

- Prise en charge du calcul des tableaux de déclaration avec des géométries multiples (Fidji, Russie)
- Ajouter les tableaux LPD et LC à l'onglet de la feuille de calcul UNCCD
- Nettoyer le message d'avertissement dans l'outil d'importation LPD
- Correction du chargement de la couche de productivité combinée finale TE
- Correction des tâches de téléchargement (toujours pas de styles)

2.36 0.44 (12 mars 2018)

- Ajouter JRC LPD
- Ajout d'un outil pour télécharger des données personnalisées de couverture du sol
- Ajout d'un outil pour le téléchargement de données de productivité personnalisées
- Ajout d'une note indiquant que le téléchargement de SOC personnalisés sera bientôt disponible
- Ajout d'un outil pour ajouter des cartes de base utilisant les données de Natural Earth
- Ajout d'un outil tout-en-un pour calculer les trois sous-indicateurs en une fois
- Renommer la classe « terres dénudées » en « autres terres » par souci de cohérence avec la CNULCD
- Mise à jour de la documentation
- Mise à jour vers marshmallow 3.0.0b7
- Déplacer le code GEE dans le dépôt principal de trends.earth
- Amélioration de la gestion des AOI, en particulier lorsque des fichiers de forme sont utilisés en entrée
- Gérer les sorties multi-fichiers de GEE en les mettant en mosaïque dans les VRT
- Prise en charge du traitement des données pour les pays qui traversent le 180e méridien
- Amélioration du formatage du tableau récapitulatif
- Dorénavant, les versions du script GEE correspondront à la version du module d'extension

2.37 0.42 (4 février 2018)

- Correction du blocage lors du changement de l'agrégation LC (dû à setEnabled sur l'étiquette supprimée)

2.38 0.40 (4 février 2018)

- Supprimer l'utilisation du mode pour l'indicateur de couverture des terres.
- Combiner le tableau récapitulatif et les outils de création de cartes d'indicateurs SDG.
- Ajouter un talon pour indiquer où le produit LPD du CCR sera disponible.
- Enregistrer le sous-indicateur de productivité comme bande 2 dans le fichier d'indicateurs SDG.
- Passage du script GEE à la v0.3.
- Correction de l'erreur due à la division par zéro lors de la génération du tableau récapitulatif lorsqu'une classe a une superficie nulle.
- Utilisation par défaut de MODIS pour les calculs de productivité.

2.39 0.38 (16 janvier 2018)

- Ajout du calcul annuel du carbone organique du sol
- Nettoyer le code de traitement des AOI, autoriser les polygones d'entrée multiples dans les AOI de fichier de forme
- Ajout d'un ombrage sur le côté des éléments du tableau d'agrégation de la couverture terrestre
- Correction du problème de firstShow sur le tableau d'agrégation
- Réviser la sortie du tableau récapitulatif pour fournir des informations supplémentaires sur chacun des trois indicateurs
- Ajouter des ensembles de données supplémentaires à la sortie de la performance, de l'état, de la couverture terrestre et du carbone organique du sol.
- Mettre à jour les valeurs de non données et de masquage pour qu'elles soient systématiquement -32768 (pas de données) et -32767 (données masquées)
- Permettre de nommer les téléchargements de fichiers
- Ajouter une icône au menu de la barre d'outils, fixer le nom du module d'extension.
- Refactoriser le code de style de couche pour tirer les informations de bande de la sortie GEE.
- Ajouter un outil pour charger les jeux de données trends.earth existants dans QGIS.
- Correction des limites de date de couverture du sol ; ne permet pas de sélectionner des dates invalides à partir des données CCI.

2.40 0.36 (14 décembre 2017)

- Correction du problème avec showEvent sur l'outil de création de rapports de carte.

2.41 0.34 (14 décembre 2017)

2.42 0.32 (14 décembre 2017)

2.43 0.30 (12 décembre 2017)

2.44 0.24 (6 décembre 2017) <<https://github.com/ConservationInternational/tren>

2.45 0.22 (4 décembre 2017) <<https://github.com/ConservationInternational/tren>

2.46 0.18 (2 décembre 2017)

2.47 0.16 (6 novembre 2017)

2.48 0.14 (25 octobre 2017) <<https://github.com/ConservationInternational/trend>

2.49 0.12 (6 octobre 2017)