

Procédure annuel pour updater les taxes (updated version 2026)

Note: La procédure assume que les liens utilisés resteront les mêmes et que la structure des fichiers PDF ne changera pas. Si les liens sont modifiés ou si la structure des PDF change, le script devra être ajusté en conséquence.

S.V.P Veuillez tout lire avant de faire la procédure

La mise à jour annuelle est maintenant automatisée avec le script :

```
node.js/exports/updateAnnualTaxData.js
```

Le script utilise les sources suivantes :

- CRA / ARC T4032 pour les tables fédérales et provinciales ON, NB, PE, QC
- Revenu Québec pour les tables provinciales QC, RRQ, RQAP
- CRA / Revenu Québec pour les taux CPP, EI, GST/HST, QST

Les sources et valeurs configurées sont dans :

```
node.js/exports/annualTaxSources.json
```

Comment ça fonctionne:

1. Copier manuellement le Google Sheet income_taxes_<année précédente>.
2. Renommer la copie en changeant l'année précédente par l'année en cours

3. Copier l’ID du nouveau Google Sheet. (L’ID est la partie dans l’URL entre /d/ et /edit.)
Exemple: https://docs.google.com/spreadsheets/d/190fSZq-r3K65Fb_g1D0oTQQZv5ijtIMjXvthD8Mlhwm/edit

4. Lancer le self-test :

```
node exports/updateAnnualTaxData.js --year 2026 --templateYear 2025 --mode full --selfTest
```

Ce test vérifie :

- que les PDFs officiels sont accessibles
- que les tables sont parsées correctement
- que chaque table commence avec la ligne `min_salary = 0`
- que les salaires minimums sont en ordre croissant
- que les tables Québec sont mappées correctement (cas spécial)
- que les zéros sont exportés au bon format pour Google Sheets

5. Lancer un dry run :

```
node exports/updateAnnualTaxData.js --year 2026 --templateYear 2025 --mode full --dryRun --incomeTaxesSpreadsheetId <SPREADSHEET_ID>
```

Le dry run ne modifie rien. Il affiche :

- les sources utilisées
- les fichiers qui seraient changés
- les onglets Google Sheets qui seraient modifiés
- les PDFs qui seraient uploadés
- les changements de taux entre l’année précédente et la nouvelle année

6. Lancer l’export réel :

```
node exports/updateAnnualTaxData.js --year 2026 --templateYear 2025 --mode full --apply --incomeTaxesSpreadsheetId <SPREADSHEET_ID>
```

Utilisation Dans VS Code

Les configurations mentionnées ci-dessus existent dans `.vscode/launch.json` :

```
Annual Tax Export - Self Test  
Annual Tax Export - Dry Run  
Annual Tax Export - Run
```

Lors du lancement, VS Code demande :

- `taxYear` : l'année à générer, ex. 2026
- `templateTaxYear` : l'année précédente à utiliser comme modèle, ex. 2025
- `incomeTaxesSpreadsheetId` : l'ID du Google Sheet copié manuellement
- `annualTaxTargets` : les destinations à mettre à jour
- `setDefaultYear` : si on veut changer l'année en cours ou non

Choix Des Destinations

Le script permet de relancer seulement une partie du processus avec `--targets`.

Options disponibles :

```
all
incomeTaxes
byModule
do
incomeTaxes,byModule
incomeTaxes,do
byModule,do
```

Signification :

`incomeTaxes`

Met à jour le Google Sheet copié `income_taxes_<année>` :

- écrit les onglets de tables d'impôt
- renomme les tabs de l'année précédente vers la nouvelle année
- met à jour `exports/exportAllData.js` avec le nouvel ID de sheet et les gids des tabs

`byModule`

Met à jour les colonnes de la nouvelle année dans le Google Sheet principal :

- `taxes`
- `income_taxes`
- `employer_income_taxes`
- `interests`

`do`

Upload les PDFs officiels dans DigitalOcean Spaces :

```
acc1/<année>/<theme>_<lang>
```

Fichiers Modifiés Par Le Script

Selon les targets choisis, le script peut modifier :

```
game/Assets/GameSpecific/byModule/Resources/data_byModule/income_tax_*_<année>.json
game/Assets/GameSpecific/byModule/Resources/data_byModule/taxes.json
game/Assets/GameSpecific/byModule/Resources/data_byModule/income_taxes.json
game/Assets/GameSpecific/byModule/Resources/data_byModule/employer_income_taxes.json
game/Assets/GameSpecific/byModule/Resources/data_byModule/interests.json
node.js/exports/exportAllData.js
```

Si `--setDefaultYear` est utilisé, il modifie aussi :

```
game/Assets/GameSpecific/byModule/Resources/data_byModule/act_settings.json
uniksimweb/uniksim-back-end/src/node.js/server/backend_braincloud.js
```

`--setDefaultYear` change l'année en cours utilisé dans la simulation.

Veuillez vous référer aux étapes ci-dessous si vous décidez de ne pas changer l'année tout de suite.

- Modifier le champs year pour l'année par défaut voulu (`act_settings.json`)

```
{
  "region": "ca_qc",
  "year": "2023",
}
```

- Modifier l'année par défaut pour la création de sims démo (`backend_braincloud.js`)

```
217 // Add region for acc1
218 if (gameUid == "acc1") {
219   let actSettingsOverride = {
220     "region": "ca_qc",
221     "year": "2024"
222   };
223   data["additionalData"] = {
224     "actSettingsOverride": actSettingsOverride
225   };
226 }
```

- Exporter les scripts acc1/acc2 et income_taxes_*

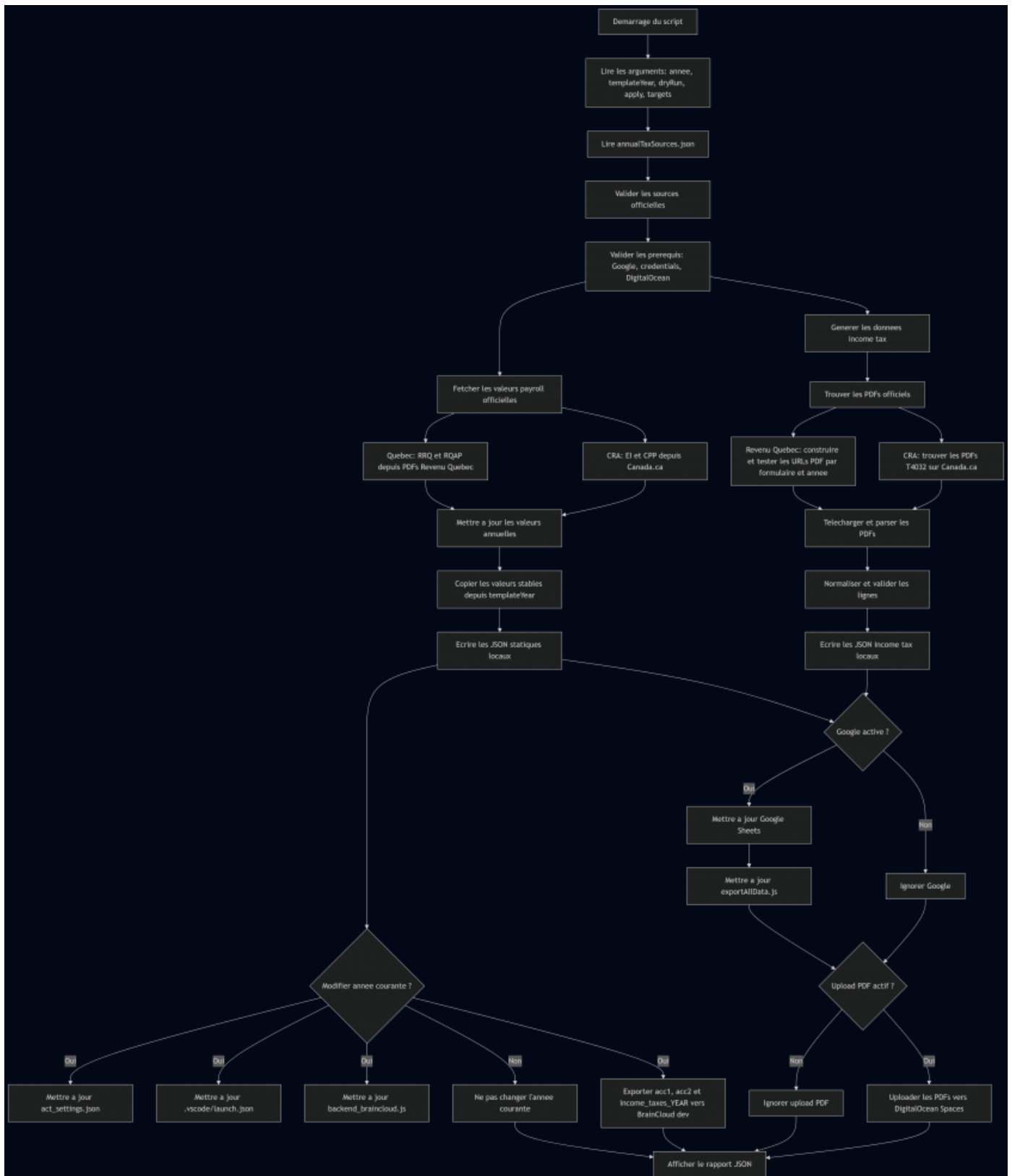
```
exporting AllData:acc1  
Ready to start exporting acc1's data to Braincloud? (y [-demo] [-i]/ n): y
```

```
exporting AllData:income_taxes_2023  
Ready to start exporting income_taxes_2023's data to Braincloud? (y [-demo] [-i]/ n): y
```

Finalement, il faut incrémenter la version dans version.txt et build pour que la nouvelle année en cours prenne effet.

```
Assets\Resources\data\version.txt
```

Diagramme qui montre comment le script fonctionne:



Revision #20

Created Thu, Jun 11, 2026 8:25 PM by [Mathis](#)

Updated Fri, Jun 12, 2026 3:53 PM by

