

Référence des fichiers de configuration et de fonctionnement de darktable

Ce document décrit les fichiers et dossiers conformes à la version 4.2 de darktable. La précédente édition mentionnait la conformité avec la version 3.8. Il n'y a pas eu de modification des fichiers de configuration pour la version 4.0, donc pas d'édition intermédiaire.

Les nouveautés de cette édition - ajouts ou suppressions - sont ==surlignées==. De plus les suppressions sont barrées. Elles subsistent ensuite dans le texte avec mention de la version de suppression.

Indépendamment de vos propres fichiers d'images, darktable utilise un certain nombre de fichiers pour fonctionner : ces fichiers sont de plusieurs ordres.

- On distinguera d'abord les fichiers qui sont propres à chaque utilisateur de darktable et ceux qui sont communs à tous les utilisateurs. Ces derniers sont fournis systématiquement avec chaque nouvelle version de darktable. Les modifications que vous feriez sur ces derniers seront donc perdues lors de chaque mise à jour. Je les référencerai comme fichier utilisateur ou système.
- On distinguera ensuite les fichiers de configuration qui régissent le fonctionnement du logiciel, des fichiers de traitement qui matérialisent toutes les modifications que vous faites sur vos photos. En effet, darktable ne modifie JAMAIS vos fichiers d'images. Toutes vos modifications sont enregistrées dans ces fichiers de traitement. Je les référencerai comme fichier de configuration, ou de développement.
- On distinguera enfin les fichiers de type "texte" qui n'ont besoin que d'un éditeur de texte pour être consultés ou modifiés, des fichiers de type "binaire" qui nécessitent des outils spécialisés pour être consultés ou modifiés. Je les référencerai également comme texte ou binaire.

Emplacement des fichiers.

Tous ces fichiers sont stockés à plusieurs endroits sur votre machine. Un endroit destiné essentiellement à vos propres fichiers que j'appellerai bien sûr le dossier **Utilisateur** ; un deuxième qui contient les vignettes de vos photos ainsi que quelques données d'exécution de darktable que j'appellerai dossier **Cache** ; un troisième qui sera utilisé pour les instantanés et (supprimé en 4.2) pour les enregistrements de débogage que j'appellerai dossier **Temporaire** ; un quatrième destiné aux fichiers système que j'appellerai dossier **Système** cela va de soi ! Et enfin tous les dossiers qui contiennent des images que vous avez importées. Ces dossiers sont situés différemment selon le système que vous utilisez. Voici ces lieux décrits système par système. Vous pouvez aussi définir vous-même l'emplacement de ces dossiers grâce aux options de lancement de darktable.

Sous Linux.

Remarque préliminaire : La variable \$HOME contient le chemin absolu (à partir de la racine du système), du dossier personnel de l'utilisateur. C'est-à-dire, en général, /home/<login de l'utilisateur>. Cette variable est utilisée couramment dans les scripts et en mode ligne de commande. Elle n'est pas universellement utilisable dans les gestionnaires de fichiers graphiques ; par exemple, Nautilus ne la reconnaît pas alors que Dolphin la reconnaît. Je l'ai employée dans cette description, car c'est une façon assez élégante de décrire cette partie du chemin d'accès à des fichiers. Nota bene : le caractère ~ remplace avantageusement la variable \$HOME en saisie ; de plus il est utilisable partout.

- Dossier Utilisateur : \$HOME/.config/darktable par défaut. Cet emplacement peut être modifié avec l'option de lancement --configdir.
- Dossier Cache : \$HOME/.cache/darktable par défaut. Cet emplacement peut être modifié avec l'option de lancement --cachedir.
- Dossier Temporaire : /tmp par défaut. Cet emplacement peut être modifié avec l'option de lancement --tmpdir.
- Dossier Système : /usr/share/darktable par défaut. Cet emplacement peut être modifié avec l'option de lancement --datadir.

Si vous compilez vous-même la version la plus récente de darktable, le dossier système se trouve par défaut dans /opt.

Le dossier .config est un dossier caché (un '.' au début du nom). Pour y accéder, il suffit de taper son nom. Voici les moyens pour le visualiser - Dans un terminal : `ls -la` - Avec les outils graphiques, cela dépend de l'outil, mais le raccourci Ctrl+h permet le plus souvent de le voir.

Sous Mac os.

Remarque préliminaire : Mac os et Linux partagent le même ancêtre commun, Unix. La remarque préliminaire concernant la variable \$HOME, héritée d'Unix, s'applique donc également ici.

- Dossier Utilisateur : \$HOME/.config/darktable par défaut. Cet emplacement peut être modifié avec l'option de lancement --configdir.
- Dossier Cache : \$HOME/.cache/darktable par défaut. Cet emplacement peut être modifié avec l'option de lancement --cachedir.
- Dossier Temporaire : /private/var/folders/h_/<hashcode>/T par défaut. Cet emplacement peut être modifié avec l'option de lancement --tmpdir.
- Dossier Système : /Applications/darktable.app/Contents/Resources/share/darktable par défaut. Cet emplacement peut être modifié avec l'option de lancement --datadir.

Le dossier .config est un dossier caché (un '.' au début du nom). Pour y accéder, il y a plusieurs méthodes selon qu'on utilise un terminal ou le

finder. Plusieurs de ces méthodes sont décrites dans ce [fil de discussion](#)

Sous Windows.

- Dossier Utilisateur : %LOCALAPPDATA%\darktable par défaut. Cet emplacement peut être modifié avec l'option de lancement `--configdir`
- Dossier Cache : L'emplacement par défaut est (était ? Changement de version de Windows ?) %LOCALAPPDATA%\Microsoft\Windows\Temporary Internet Files\darktable\ Je l'ai vu aussi ici (version de Windows ?) : %LOCALAPPDATA%\Microsoft\Windows\NetCache\darktable\ Cet emplacement peut être modifié avec l'option de lancement `--cachedir`.
- Dossier Temporaire : %LOCALAPPDATA%\Temp\ par défaut. Cet emplacement peut être modifié avec l'option de lancement `--tempdir`.
- Dossier Système : C:\Program Files\darktable\share\darktable\ par défaut. Cet emplacement peut être modifié avec l'option de lancement `-datadir`.

La variable %LOCALAPPDATA% correspond au chemin des fichiers de configuration des applications propres à chaque l'utilisateur. C'est-à-dire, en général, C:\Users\<login de l'utilisateur>\AppData\Local\.

Comme pour Linux et Mac os, j'ai utilisé la notation visible dans un terminal. Donc "Users" et "Program Files". Ces noms apparaîtront sous une forme francisée dans l'explorateur de fichier : "Utilisateurs" et "Programmes". Le dossier AppData est un dossier caché. Pour l'afficher, passer par les options de l'explorateur : organiser → options des dossiers et de recherche → affichage → afficher les fichiers, dossiers et lecteurs cachés.

Description des fichiers.

Fichiers du dossier Utilisateur.

Voici la liste des fichiers contenus dans ce dossier avec la description de leur rôle. Ils ne sont pas obligatoirement tous présents sur votre installation, car ils se créent au fil du temps et des besoins ; parfois même, leur création nécessite une intervention manuelle.

- `color` Dossier utilisateur contenant les éventuels profils de couleur d'entrée et/ou de sortie. Ce dossier ne contient pas directement les profils. Il contient uniquement deux dossiers `in` et `out`. Les dossiers `color`, `color/in` et `color/out` ne sont pas créés automatiquement lors de la configuration initiale de darktable. Si vous avez besoin de ces dossiers pour y déposer des profils, vous devez les créer.
 - `color/in` Dossier utilisateur contenant les profils de couleur d'entrée. Ces fichiers de profil sont au format ICC.
 - `color/out` Dossier utilisateur contenant les profils de couleur de sortie. Ces fichiers de profil sont au format ICC.
- `darktable.css` | Utilisateur | Configuration | Texte | Fichier contenant les modifications personnelles de l'aspect de l'interface de darktable. Ce fichier est supprimé à partir de la version 3.0 et remplacé par les fichiers du dossier `themes`.
- `darktable.rc` | Utilisateur | Configuration | Texte | Fichier contenant les données de configuration du logiciel lui-même. La plupart des données de configuration contenues dans ce fichier peuvent être modifiées via les différents onglets de la fenêtre "préférences" de darktable. Il existe cependant quelques valeurs très peu utilisées ou « un peu trop sensibles » qui ne sont manipulables que par la modification de ce fichier. Darktable enregistre ce fichier à chaque sortie du logiciel. Les éventuelles personnalisations doivent donc se faire quand darktable n'est pas actif.
- `data.db` | Utilisateur | Développement | Binaire | Fichier qui contient la seconde partie de la Base de Données (BdD). Contient les tags, les styles, les pré-réglages, la géolocalisation. Fichier au format `sqlite`. L'emplacement de ce fichier peut être modifié avec l'option de lancement `-library`. L'utilisation de cette option modifiera simultanément l'emplacement de `data.db` et de `library.db`.
- `data.db.lock` | Utilisateur | Configuration | Texte | Fichier de verrouillage de la seconde partie de la BdD. Il est créé au moment de l'ouverture de celle-ci, et supprimé lors de sa fermeture. Dans de rares cas, darktable ne peut pas supprimer ce fichier avant de s'arrêter. S'il subsiste, le prochain lancement de darktable avortera puisque celui-ci considérera que la BdD est déjà ouverte par un autre logiciel. Dans ce cas, sa suppression réglera le problème. Il contient le "Process Id" du processus qui a ouvert la BdD. Ce fichier suit le fichier `data.db`. Si l'emplacement de `data.db` a été modifié avec l'option `-library`, l'emplacement du fichier de verrouillage le sera également.
- `data.db-pre<version>` | Utilisateur | Développement | Binaire | L'installation d'une nouvelle version majeure de darktable nécessite parfois (souvent) une modification du schéma de la base de données rendant incompatible la BdD avec la version antérieure. Avant cette modification, une sauvegarde est créée automatiquement par copie du fichier `data.db`. Ce fichier peut être supprimé s'il n'est pas envisagé de retour vers une version antérieure. Dans le nom, correspond au numéro de la nouvelle version de darktable en cours d'installation.
- `data.db-snp<date>` | Utilisateur | Développement | Binaire | Ces fichiers contiennent des instantanés (snapshots en anglais, d'où l'abréviation `snp`), de la BdD. C'est-à-dire, une sauvegarde de la BdD faite automatiquement et à cadence régulière, afin de pouvoir revenir à une situation correspondant à la date mentionnée dans le nom du fichier. La date est enregistrée sous forme compacte - sans séparateurs -, selon la séquence suivante `AAAAMMJJhhmmss`. Vous pouvez régler les paramètres de maintenance de la BdD dans la fenêtre des préférences de darktable → onglet stockage → paragraphe base de données. Fichier au format `sqlite`.
- `keyboardrc` | Utilisateur | Configuration | Texte |

Fichier qui contient les raccourcis, éventuellement personnalisés. Les raccourcis peuvent tous être modifiés via l'onglet "raccourcis" de la fenêtre "préférences" de darktable. Ce fichier est supprimé à partir de la version 3.8 et remplacé par le fichier `shortcutsrc`.

- `keyboardrc_default` | Utilisateur | Configuration | Texte |
Fichier qui contient les valeurs par défaut des raccourcis. C'est le seul fichier système qui est stocké dans le dossier utilisateur. Ce fichier est supprimé à partir de la version 3.8 et remplacé par le fichier `shortcutsrc.default`.
- `library.db` | Utilisateur | Développement | Binaire |
Fichier qui contient la première partie de la BdD. Contient toutes les données de développement. Fichier au format sqlite. L'emplacement de ce fichier peut être modifié avec l'option de lancement `-library`. L'utilisation de cette option modifiera simultanément l'emplacement de `data.db` et de `library.db`.
- `library.db.lock` | Utilisateur | Configuration | Texte |
Fichier de verrouillage de la première partie de la BdD. Voir `data.db.lock` ci-dessus. Si l'emplacement de `library.db` a été modifié avec l'option `-library`, l'emplacement du fichier de verrouillage le sera également.
- `library.db-pre<version>` | Utilisateur | Développement | Binaire |
L'installation d'une nouvelle version majeure de darktable nécessite souvent une modification du schéma de la base de données rendant incompatible la BdD avec la version antérieure. Avant cette modification, une sauvegarde est créée automatiquement par copie du fichier `library.db`. Ce fichier peut être supprimé s'il n'est pas envisagé de retour vers une version antérieure. Dans le nom, `<version>` correspond au numéro de la nouvelle version de darktable en cours d'installation.
- `library.db-snp-<date>` | Utilisateur | Développement | Binaire |
Ces fichiers contiennent des instantanés (snapshots en anglais, d'où l'abréviation `snp`), de la BdD. C'est-à-dire, une sauvegarde de la BdD faite automatiquement et à cadence régulière, afin de pouvoir revenir à une situation correspondant à la date mentionnée dans le nom du fichier. La date est enregistrée sous forme compacte - sans séparateurs -, selon la séquence suivante AAAAMMJJhhmmss. Vous pouvez régler les paramètres de maintenance de la BdD dans la fenêtre des préférences de darktable → onglet stockage → paragraphe base de données. Fichier au format sqlite.
- `lua` Dossier utilisateur contenant les éventuels scripts add-on (écrits en langage Lua), ceux que vous avez importés comme ceux que vous avez éventuellement créés. Les scripts sont écrits en langage Lua. Pour découvrir Lua, outre le site lua.org, vous disposez d'un chapitre dans le manuel de darktable ainsi qu'un manuel de référence dédié sur le site darktable.org. Le dossier `lua` n'est pas créé automatiquement lors de la configuration initiale de darktable. Si vous avez besoin de ce dossier pour y déposer des scripts, vous devez le créer.
- `presets.json` | Utilisateur | Configuration | Texte |
Fichier contenant les profils de bruits calculés par l'utilisateur en vue d'effectuer des tests. Ce fichier n'existe que si vous l'avez copié vous-même à cet endroit. Il est le résultat du calcul du profil de bruit d'un boîtier. Ce fichier est au format json.
- `luarc` | Utilisateur | Configuration | Texte |
Fichier qui contient la liste des scripts Lua à charger lors du lancement de darktable. Il n'y a pas d'outil pour modifier ce fichier dans darktable. Il faut le modifier vous-même avec un éditeur de texte, en respectant bien la syntaxe. Il doit être écrit en langage Lua. Voir des exemples dans le manuel de référence Lua disponible ainsi que le manuel général sur le site darktable.org
- `shortcutsrc` | Utilisateur | Configuration | Texte |
Fichier qui contient les raccourcis, éventuellement personnalisés. Les raccourcis peuvent tous être modifiés via l'onglet "raccourcis" de la fenêtre "préférences" de darktable ou via la fenêtre dédiée accessible via l'icône représentant un clavier.
- `shortcutsrc.backup` | Utilisateur | Configuration | Texte |
Fichier qui contient les valeurs sauvegardées des raccourcis. Lors d'une modification de ceux-ci.
- `shortcutsrc.default` | Utilisateur | Configuration | Texte |
Fichier qui contient les valeurs par défaut des raccourcis. C'est le seul fichier système qui est stocké dans le dossier utilisateur.
- `shortcutsrc.edit` | Utilisateur | Configuration | Texte |
Fichier utilisé par la boîte de dialogue « restauration raccourci »
- `styles` Dossier utilisateur contenant les fichiers de styles que vous avez créés. La suppression d'un de vos styles dans darktable ne supprime pas le fichier dans ce répertoire. Le dossier `styles` n'est pas créé automatiquement lors de la configuration initiale de darktable. Il est créé par darktable quand vous créez un premier style dans cette configuration. Il n'est pas nécessaire d'y déposer les styles que vous importez.
 - `<vos styles>.dtstyle` | Utilisateur | Développement | Texte |
Les fichiers du dossier `styles` sont des fichiers au format xml, mais le contenu, bien que de format texte, est plutôt un BLOB (Binary Large Object), donc non modifiable. Ces fichiers sont suffixés avec l'extension `".dtstyle"`.
- `themes` Dossier utilisateur contenant les fichiers éventuels de configuration personnalisés de l'aspect de l'interface de darktable. Ces fichiers sont au format Cascading Style Sheet (CSS) tel que décrit par le W3C. Cependant, toutes les fonctionnalités CSS ne sont pas prises en charge par l'interface graphique Gtk utilisée par darktable. Si vous avez besoin de ce dossier pour y déposer vos propres feuilles de styles, vous devez le créer.
 - `<vos thèmes>.css` | Utilisateur | Configuration | Texte |

Fichier(s) CSS. Le nommage est libre, mais il est recommandé d'utiliser un nom de fichier différent des fichiers de style fournis par défaut (voir ci-dessous le dossier thèmes dans le paragraphe 'Fichiers du dossier S'). Si vous complétez ou modifiez un de ces fichiers, ne le copiez pas en totalité. Créez votre fichier, commencez-le par une instruction d'importation telle que décrite dans la norme CSS, puis ajoutez simplement vos modifications.

- `user.css` | Utilisateur | Configuration | Texte |
Fichier CSS. Ce fichier contient les personnalisations des feuilles de style que vous faites à partir de la fenêtre des préférences de darktable → onglet général → option « modifier le thème via les ajustements CSS ci-dessous » et la zone de saisie située juste en dessous.
- `watermarks` Dossier utilisateur contenant les fichiers des filigranes. Ces fichiers doivent être au format SVG. Ils peuvent être créés avec n'importe quel outil capable de produire ce type de format de fichier. Le dossier `watermarks` n'est pas créé automatiquement lors de la configuration initiale de darktable. Si vous avez besoin de ce dossier pour y déposer des fichiers de filigranes, vous devez le créer.

Parmi ces fichiers, deux fichiers constituent la BdD ; ce qu'ailleurs on appelle le catalogue. Ces deux fichiers au format sqlite sont :

- `library.db` : qui contient toutes les données de développement. En fait, tout ce que l'on retrouve dans les `.xmp` individuels.
- `data.db` : qui contient tout le reste, les tags, les styles, les pré-réglages, la géolocalisation.

La scission en deux de la BdD a été faite pour pouvoir la régénérer à partir des fichiers XMP sans perdre les mots clés, les styles et les pré-réglages. Comme vous pouvez le constater, la plupart de ces fichiers contiennent des données qui vous sont propres. Et même, si la plupart d'entre eux peuvent être reconstitués facilement, il est certainement plus facile d'avoir une sauvegarde du tout. Donc sauvegarder le dossier complet n'est pas une mauvaise idée. D'autant plus que ces fichiers ne représentent qu'un tout petit volume d'informations. L'essentiel en volume ce sont les deux fichiers de la BdD.

À la lecture de ce qui précède, vous pouvez imaginer qu'une réinitialisation partielle ou complète de la configuration peut se faire en supprimant une partie ou la totalité des fichiers utilisateur. Attention cependant, car il existe un lien entre certaines informations contenues dans la base de donnée et les vignettes stockées dans le dossier Cache (voir paragraphe ci-dessous). Si vous supprimez la BdD, il vous faudra également réinitialiser le cache des vignettes.

Fichiers du dossier Cache.

Voici la liste des dossiers et sous dossiers du dossier cache, ainsi que la description des fichiers qu'ils contiennent

- `cached_kernels_for_<référence de votre carte graphique>` Dossier contenant les versions compilées à la volée par darktable des kernels openCL (Voir dans la section Système ci dessous, le dossier `kernels`). Si vous avez plusieurs cartes graphiques, vous aurez plusieurs dossiers de ce type, un par carte graphique.
- `mipmaps-<mipmaps Id>.d` Dossier contenant le cache des vignettes utilisées en table lumineuse et dans les bandeaux. Le `mipmaps Id` est un hashcode calculé lors de l'initialisation d'une instance de darktable. Ce dossier contient lui-même plusieurs sous dossiers nommés de 0 à 8. Lors d'une première utilisation, seuls les dossiers 0, 1 et 2 sont créés. Les autres dossiers seront créés à la demande et de façon transparente par darktable ou explicitement lors de l'utilisation de l'outil `darktable-generate-cache`. Chaque dossier correspond à une taille de vignette ; de la plus petite taille dans le dossier 0, à la plus grande dans le dossier 8. Les tailles des vignettes sont de 110, 225 et 450 pixels sur le plus grand coté pour les dossiers 0, 1 et 2.

Les vignettes elles-mêmes sont des fichiers JPEGs qui sont nommés selon le numéro de photo. Ce numéro est créé lors de l'importation, de manière totalement automatique et désigne univoquement la photo dans la BdD de darktable. Il est visible dans le module informations de l'image, à la ligne "numéro".

Attention ! Si vous réinitialisez la BdD de darktable, il faudra également réinitialiser le cache des vignettes, car comme dit ci-dessus, celles-ci sont nommées avec le numéro de la photo. Après réinitialisation de la BdD, il sera donc certain que les numéros des anciennes vignettes ne correspondront plus avec les numéros des nouvelles photos importées. L'affichage en table lumineuse sera très étrange !

Fichiers du dossier Temporaire.

Voici la description des fichiers qu'il peut contenir.

- `==dt_snapshot_<numéro>.png==` | Utilisateur | Développement | Binaire |
Fichier PNG, qui contient l'état d'une photo en cours de développement et permet d'afficher les instantanés dans la chambre noire. Le numéro permet de différencier les différents instantanés. Il ne correspond pas à la pile d'historique. Par défaut, il démarre à trois et décroît jusqu'à zéro. Il n'est de ce fait pas possible de disposer de plus de quatre instantanés simultanément. Si vous en créez un cinquième, le plus ancien sera perdu. Quand vous quittez darktable, ces fichiers ne sont pas supprimés ; ils subsistent dans le dossier temporaire. Contrairement au dossier du cache des vignettes (voir ci-dessus dans la description du dossier temporaire), qui est composé du nom `mipmaps` et d'un hashcode, aucun élément des noms de fichiers d'instantanés ne permet d'associer un de ces fichiers à une occurrence de darktable. Il y a donc risque de conflit, en particulier, si les occurrences appartiennent à deux utilisateurs différents.. Ces fichiers ne sont plus produits à partir de la version 4.2 Les instantanés sont uniquement conservés en mémoire.
- `darktable_bt_<numéro>.txt` | Utilisateur | Développement | Texte |
Fichier de débogage. Ce fichier peut être produit par darktable en cas de plantage. Il pourra être utile aux développeurs pour identifier la partie du code à l'origine du plantage. Dans ce cas, cependant, il sera nécessaire de reproduire le plantage avec une version de darktable compilée afin que les "symboles" soient inclus dans ce fichier. Le numéro est variable à chaque lancement de darktable (probablement le numéro de processus, mais je n'ai pas pu le vérifier car je n'ai pas de plantages ;-)).

Fichiers du dossier système.

Voici la liste des fichiers de ce dossier avec la description de leur contenu.

- `darktable.bash` | Système | Configuration | Texte |
Fichier de lancement de darktable. [à compléter !]
- `darktable.css` | Système | Configuration | Texte |
Fichier de configuration de l'aspect de l'interface de darktable. Ce fichier est supprimé à partir de la version 3.0 et remplacé par un fichier équivalent dans le dossier `themes`.
- `darktable.rc` | Système | Configuration | Texte |
Fichier contenant les données de configuration du logiciel lui-même. Il s'agit du fichier par défaut qui est copié lors du premier démarrage d'une nouvelle instance de darktable.
- `gdb_commands` | Système | Configuration | Texte |
Fichier contenant quelques commandes utilisées lors d'un lancement de darktable via la commande `gdb` à des fins de debug.
- `kernels` | Système | Configuration | Texte |
Dossier contenant les codes support OpenCL des modules correspondants. Ces fichiers sont au format langage "c".
- `latex` Dossier contenant un fichier de configuration utilisé lors de l'exportation sous forme de "modèle de livre LaTeX" (langage LaTeX)
- `lua` Dossier contenant un fichier utilisé pour le debug des scripts Lua (langage Lua)
- `luarc` | Système | Configuration | Texte |
Fichier initial de Lua.
- `noiseprofiles.json` | Système | Configuration | Texte |
Fichier contenant tous les profils de bruits connus de darktable Fichier au format json.
- `pixmaps` Dossier contenant les icônes utilisées en interne par darktable.
- `pswp` Dossier contenant les fichiers nécessaires pour l'exportation sous forme de "galerie web". Voir également le dossier `style`.
- `rawspeed` Dossier contenant les fichiers de configuration de Rawspeed. Rawspeed est la librairie sous-jacente de darktable qui décode les fichiers raw. Ce dossier contient deux fichiers :
 - `cameras.xml` | Système | Développement | Texte |
Ce fichier XML décrit tous les boîtiers pris en charge par rawspeed et donc par darktable. Il contient sa propre DTD. Chaque entrée décrit toutes les caractéristiques du capteur du boîtier.
 - `showcameras.xsl` | Système | Développement | Texte |
Ce second fichier est la feuille de style nécessaire pour imprimer le fichier `cameras.xml`, le cas échéant,
- `style` Dossier contenant les feuilles de style (css), ainsi que quelques fichiers jpeg, gif et ico, nécessaires pour l'exportation sous forme de "galerie web". Voir aussi le dossier `pswp`.
- `themes` Dossier contenant les fichiers de configuration de l'aspect de l'interface de darktable. Ces fichiers sont au format Cascading Style Sheet (CSS) tel que décrit par le W3C. Cependant, toutes les fonctionnalités CSS ne sont pas prises en charge par l'interface graphique Gtk utilisée par darktable. Ce dossier contient les fichiers suivants. Tous sont classifiés
| Système | Configuration | Texte |
 - `darktable.css` Feuille de style d'origine de darktable.
 - `darktable-elegant-dark.css` Nouvelle feuille de style sombre.
 - `darktable-elegant-darker.css` Nouvelle feuille de style plus sombre.
 - `darktable-elegant-grey.css` Nouvelle feuille de style claire.
 - `darktable-icons.css` Nouvelle feuille de style sombre avec icônes.
 - `darktable-icons-dark.css` Nouvelle feuille de style sombre avec icônes.
 - `darktable-icons-darker.css` Nouvelle feuille de style plus sombre avec icônes.
 - `darktable-icons-grey.css` Nouvelle feuille de style claire avec icônes.
- `tools` Dossier qui contient quelques programmes utilitaires d'administration de darktable. Actuellement ce sont des scripts bash. En voici la liste :
 - `common.sh`
 - `extract_wb_from_images.sh`
 - `purge_from_cache.sh`
 - `purge_non_existing_images.sh`
 - `purge_unused_tags.sh` Attention ! Ces scripts sont disponibles dans la version Windows, mais pas utilisables tels quels. Ce sont

des scripts en langage bash qui nécessitent l'interpréteur bash, non disponible nativement sous windows.

- **watermarks** Dossier contenant les quelques fichiers de filigrane disponibles par défaut avec darktable. Ces fichiers sont au format SVG (sur une base xml). En voici la liste. Tous sont classifiés

| Système | Développement | Texte |

- **darktable.svg**
- **hasselblad.svg**
- **promo.svg**
- **simple-text.svg** Vous pouvez copier ces fichiers dans votre propre dossier de configuration afin de les utiliser comme base de travail et créer vos propres filigranes.

- **==wb_presets.json==** | Système | Configuration | Texte |

Fichier contenant tous les préréglages de balance des blancs des appareils photos connus de de darktable Fichier au format json.

Vous avez peut-être remarqué que je n'ai pas parlé des fichiers de caractérisation des objectifs. Il y a plusieurs raisons à cela : - Ces fichiers n'appartiennent pas à darktable. Ils dépendent d'une librairie tierce : Lensfun. - Puisqu'ils appartiennent à Lensfun, ils sont fournis par cette librairie ; pas par darktable. - darktable utilise plusieurs versions différentes de Lensfun, selon les distributions et / ou les packages. - L'emplacement des fichiers dépend de la version de Lensfun, et du mode de mise à jour de ces fichiers ; parfois aussi de la distribution. - L'entête des fichiers de Lensfun change légèrement selon la version.

Bref tout cela fait beaucoup de variables et il faudrait probablement un document aussi volumineux que celui-ci pour faire l'inventaire exhaustif de toutes les hypothèses. Ce sujet a déjà été traité abondamment sur le forum darktable.fr, sans que quelques règles simples puissent être établies. Donc, pour l'instant, je n'en parle pas.

Fichiers dans vos dossiers de photos.

| Utilisateur | Développement | Texte |

En plus de tous ces fichiers de configuration ou de fonctionnement, darktable produit des fichiers XMP qui sont associés à chaque fichier image que vous avez importé dans darktable. Ces fichiers sont nommés en ajoutant l'extension ".xmp" au nom complet de vos fichiers images (nom + extension). Ces fichiers contiennent toutes les données nécessaires pour pouvoir reproduire le traitement de vos images, même si vous avez supprimé l'intégralité des fichiers que j'ai listés ci-dessus. Tant que vous avez ces fichiers, vous n'avez rien perdu. Ces fichiers sont au format XMP (sur une base xml). La norme XMP est maintenant une norme ISO sous le numéro ISO-16684-1:2012

Voilà, j'espère ne pas en avoir oublié. N'hésitez pas à me signaler toute erreur ou omission !

Jean-Pierre Verrue (contact at jpverrue point fr)