
MusicBrainz Picard

Version v2.6.3

août 20, 2021

MusicBrainz Picard User Guide by Bob Swift is licensed under CC0 1.0. To view a copy of this license, visit <https://creativecommons.org/publicdomain/zero/1.0>

Contents

1	Introduction	1
1.1	Picard peut...	2
1.2	Picard ne peut pas ...	2
1.3	Limites	3
2	Contribuer au projet	4
3	Remerciements	5
3.1	Éditeur et responsable de la langue anglaise	5
3.2	Équipes de traduction	5
3.3	Les contributeurs	6
4	Glossaire des termes	7
5	Commencer	11
5.1	Téléchargez et installez Picard	11
5.2	Écran principal	14
5.3	Icônes d'état	20
6	Configuration	23
6.1	Configuration de l'écran	23
6.2	Options d'action	24
6.3	Paramètres des options	24
7	Balises & Variables	72
7.1	Balises de base	73
7.2	Balises avancées	78
7.3	Variables de base	80
7.4	Variables de fichier	82
7.5	Variables avancées	83
7.6	Balises de musique classique	83
7.7	Balises des plugins	84
7.8	Autres informations	87

8 Script	89
8.1 Syntaxe	89
8.2 Variables de métadonnées	90
9 Fonctions de script	91
9.1 Fonctions d'affectation	91
9.2 Fonctions de texte	95
9.3 Fonctions à valeurs multiples	107
9.4 Fonctions mathématiques	115
9.5 Fonctions conditionnelles	118
9.6 Fonctions d'information	129
9.7 Fonctions de boucle	130
9.8 Fonctions diverses	131
10 Utiliser Picard	133
10.1 Récupération des informations sur l'album	133
10.2 Mise en correspondance des fichiers aux pistes	147
10.3 Définition de la pochette	149
10.4 Enregistrement des fichiers mis à jour	151
11 Recommandations de flux de travail	153
11.1 Lorsque le CD est disponible	154
11.2 Lorsque les fichiers sont regroupés par album	155
11.3 Lorsque les fichiers ne sont pas groupés mais ont des métadonnées	156
11.4 Lorsque les fichiers ne sont pas regroupés et ont peu ou pas de métadonnées existantes	157
12 Autres tâches de Picard	159
12.1 Joindre un ID de disque à une version	159
12.2 Soumission d'empreintes acoustiques	162
12.3 Générer des balises à partir des noms de fichiers	166
13 Prolonger Picard	169
13.1 Plugins	169
13.2 Scripts	171
13.3 Ordre de traitement	175
14 Dépannage	177
14.1 Dépannage général	177
14.2 Picard ne démarre pas	178
14.3 Il n'y a pas de pochette	180
14.4 Les balises ne sont pas mises à jour ou enregistrées	181
14.5 Les fichiers ne sont pas enregistrés	182
14.6 Picard vient de cesser de travailler	183
14.7 macOS montre que l'application est endommagée	183
15 Questions fréquemment posées	185
15.1 Utiliser Picard	185
15.2 Formats de fichiers	187

15.3 Configuration	190
16 Tutoriels	192
16.1 Ecriture d'un script de nommage de fichier	192
16.2 Compréhension Empreintes acoustiques et AcoustIDs	195
16.3 Gestion de plusieurs pays de sortie.	197
16.4 Écriture d'un Plugin	199
16.5 Chargement des versions avec MusicBrainz pour Android	204
17 Annexes	217
17.1 Annexe A: API des plugins	217
17.2 Annexe B: Mappage de balises	223
17.3 Annexe C: Options de ligne de commande	246
17.4 Annexe D: Courts clavier	248

CHAPTER 1

Introduction

MusicBrainz Picard est un tagueur de fichiers musicaux multiplateforme. Pour toutes les personnes qui ne savent pas ce que cela signifie, voici une explication rapide qui peut être ignorée par les personnes qui le savent déjà.

Vos fichiers musicaux ne contiennent pas que de la musique. Ils contiennent également des « métadonnées », constituées de « tags » qui se composent d'un nom ou d'un type de tag et des données associées, par exemple le nom de l'album ou de la piste, le nom de l'artiste, le label, l'année d'émission, etc. Sauf si vous extrayez vous-même les fichiers musicaux avec un outil très basique, vos fichiers musicaux contiennent probablement déjà des métadonnées de base, mais il y a littéralement des centaines de balises qui peuvent être appliquées à votre musique si vous êtes intéressé.

Évidemment, si vous le souhaitez, vous pouvez rechercher minutieusement toutes ces informations pour chaque album et suivre vous-même, et saisir les données dans un outil de marquage. Cependant, à l'ère de la connexion Internet, il est clairement plus logique qu'une personne fasse cela pour chaque album et piste, puis télécharge ces données dans une base de données partagée. L'outil de marquage peut alors accéder à cette base de données et utiliser les données pour marquer les fichiers musicaux. Et **c'est** ce que fait MusicBrainz Picard.

MusicBrainz est la base de données et **Picard** est l'outil qui balise les fichiers musicaux.

Ce guide de l'utilisateur a pour but de fournir des informations complètes sur l'utilisation de **MusicBrainz Picard** et de le rendre également disponible dans d'autres formats, y compris une version PDF adaptée à l'impression. Des liens vers des informations supplémentaires telles que des scripts, des plugins et des didacticiels sont fournis lorsqu'ils sont disponibles plutôt que d'essayer de reproduire les informations de ce document.

Afin d'utiliser efficacement Picard, il est important de comprendre ce que le programme peut faire et, tout aussi important, ce qu'il ne peut pas faire. Picard est principalement destiné à étiqueter et organiser les albums contenant des pistes, guidé par l'utilisateur vers la sortie spécifique de l'album dont il dispose, puis à maintenir les métadonnées de ces pistes à jour à mesure que les utilisateurs du monde entier améliorent la qualité de MusicBrainz. les données associées à cette version et à cette piste particulières; Picard le fait très bien en effet. Cependant, il n'est pas destiné à organiser automatiquement votre collection de milliers de fichiers musicaux aléatoires, et si c'est ce que vous espérez, vous serez probablement déçu. Pour citer le site Web de Picard, « *Picard n'est pas conçu pour être un fixateur d'étiquettes à piste unique de masse. Picard croit en la qualité plutôt que la quantité et propose une pléthora de personnalisations pour adapter les collections de musique à vos besoins.* »

1.1 Picard peut...

... ajouter des balises de métadonnées à vos fichiers musicaux, en fonction des informations disponibles sur le [site Web MusicBrainz](#).

... recherchez les métadonnées manuellement ou automatiquement en fonction des informations existantes, notamment le nom de l'artiste et du morceau, l'identifiant du disque (pour les CD) et l'empreinte digitale AcoustID d'une piste.

... récupérez et intégrez des images de pochettes à partir d'une variété de sources.

... renommez et placez les fichiers musicaux dans des répertoires en fonction des instructions de modèle de dénomination fournies dans un script de dénomination.

... calculer et soumettre un identifiant de disque à la base de données MusicBrainz, en le joignant à une version spécifiée.

... calculer et soumettre l'empreinte digitale AcoustID d'un fichier musical à la [base de données AcoustID](#).

1.2 Picard ne peut pas ...

... identifier et supprimer automatiquement tous les fichiers musicaux en double de votre collection.

... fournir des métadonnées qui n'existent pas déjà dans la base de données MusicBrainz.

1.3 Limites

Formats de fichier

Picard prend actuellement en charge la plupart des formats de fichiers musicaux, Matroska (.mka) étant une exception notable. Les fichiers Microsoft WAVE (.wav) peuvent être empreintes digitales et renommés et peuvent être marqués à l'aide de balises ID3v2, mais cela n'est pas pris en charge par tous les logiciels de lecture. De plus, Picard ne prend pas en charge l'écriture de balises personnalisées pour tous les formats.

La section *Picard Tag Mappings* fournit plus d'informations concernant le mappage entre les noms de balises internes Picard et divers formats de balisage.

Demande de limitation de débit

La récupération des métadonnées de Picard est limitée à la limitation de débit standard **une demande par seconde** pour l'API MusicBrainz. Cela devient tout à fait visible lorsque vous essayez de traiter une grande liste de fichiers, et est exacerbé par les extensions qui effectuent des demandes d'informations supplémentaires à partir de la base de données.

Traitement de fichiers réseau

Parfois, Picard doit réécrire l'intégralité du fichier de musique afin d'ajouter ou de mettre à jour les balises. Cela peut prendre quelques secondes et le délai devient encore plus long si le fichier est accédé via un réseau (par exemple: le fichier est lu ou écrit sur un périphérique NAS). La « meilleure pratique » recommandée consiste à traiter tous les fichiers sur un lecteur local, puis à les déplacer vers le répertoire distant souhaité une fois le traitement terminé.

CHAPTER 2

Contribuer au projet

Ce document n'existe que grâce à l'effort bénévole qui a été consacré à son développement, à partir de la documentation initiale sur le site Picard, des informations publiées dans le forum de discussion de la communauté, de la documentation des scripts, des plugins et du code source du programme, des relecteurs, des éditeurs, des traducteurs et les commentaires de la communauté des utilisateurs.

D'autres contributions de haute qualité sont les bienvenues de la part de tous les utilisateurs Picard souhaitant faire partie de la communauté open source qui crée et maintient cet outil musical précieux. Même si vous ne pouvez pas écrire de code, sur la base de votre expérience d'utilisation de Picard, toute aide que vous pouvez apporter pour améliorer cette documentation sera très appréciée. Même si vous ne pouvez pas améliorer l'aide existante, si vous pouvez créer ou maintenir des traductions dans d'autres langues, ce serait un grand avantage.

Si vous remarquez une erreur dans la documentation ou avez du matériel supplémentaire à contribuer, veuillez soulever un problème sur le [projet de documentation sur GitHub](#). Les demandes d'extraction pour résoudre les problèmes en suspens sont également appréciées.

Voir aussi:

[Contribuer à MusicBrainz Picard / Traductions Picard / Contribuer à la documentation](#)

CHAPTER 3

Remerciements

Nous remercions chaleureusement les personnes suivantes pour leur contribution au développement, au maintien et à l'amélioration du guide de l'utilisateur Picard:

3.1 Éditeur et responsable de la langue anglaise

- Bob Swift (rdswift)

3.2 Équipes de traduction

3.2.1 Français (Lead: TBA)

- TBA

3.2.2 Allemande (Lead: TBA)

- TBA

3.2.3 Espagnol (Lead: TBA)

- TBA

3.3 Les contributeurs

Cette liste comprend les contributeurs à la documentation, quel que soit leur rôle. En plus des contributeurs de contenu réels, cela inclut les prospects, les traducteurs, les réviseurs et les correcteurs. Toutes les contributions sont précieuses et appréciées. À ce titre, les noms sont présentés par ordre alphabétique afin de ne pas impliquer l'importance relative de la contribution d'une personne en fonction de sa position dans la liste.

Les contributeurs comprennent (par ordre alphabétique par nom de famille):

- Vedant Chakravadhanula
- Pavan Chander
- Rahul Kumar Gupta
- Wieland Hoffmann
- jesus2099
- David Kellner
- Jun Kim
- Sambhav Kothari
- Laurent Monin
- Akash Nagaraj
- Frederik « Freso » S. Olesen
- Theodore Fabian Rudy
- Sophist
- Bob Swift
- Akshat Tiwari
- Philipp Wolfer
- Shadab Zafar

Note: Il y en a probablement d'autres qui n'ont pas encore été identifiés, nous nous excusons donc si votre nom a été oublié. Veuillez nous le faire savoir et nous veillerons à ce qu'il soit ajouté à la liste.

CHAPTER 4

Glossaire des termes

De nombreux termes utilisés dans cette documentation et dans Picard lui-même ont une signification spécifique dans l'environnement MusicBrainz. Les termes spécifiques sont définis comme suit:

acoustic fingerprint

Une empreinte acoustique est un résumé numérique d'un signal audio, qui peut être utilisé pour identifier rapidement l'audio.

Veuillez consulter [Wikipedia](#) pour une explication complète de l'empreinte digitale acoustique.

AcoustID

AcoustID est un système d'empreinte digitale acoustiques entièrement basé sur une technologie open source. Consultez le site [Web AcoustID](#) pour plus d'informations.

albumartist

Le musicien ou groupe de musiciens jouant sur un album. Par exemple, « The Beatles » est le albumartist de l'album « [Past Masters, Volume One](#) », tandis que l'albumartist de « [Pas de frontières: un avantage pour les réfugiés kosovars](#) » est « Various Artists » .

Note: L'utilisation de albumartist est différente pour les versions de musique classique, qui suivent le MusicBrainz [Classical Style Guide](#), listant d'abord les compositeur, suivi par les interprètes.

artist

Le musicien ou le groupe de musiciens jouant sur une piste. Par exemple, « Jeen » est l'artiste sur la piste « Be (One in a Million) » sur l'album « Tourisme ».

Veuillez consulter la page [Artiste](#) sur le site Web de MusicBrainz pour plus d'informations.

Note: L'utilisation de l'artiste est différente pour les versions de musique classique, qui suivent le guide de style classique MusicBrainz, ne montrant que le compositeur et non les interprètes.

artist credit

Les crédits d'artiste indiquent qui est le ou les principaux artistes crédités pour les sorties, les groupes de sortie, les pistes et les enregistrements, et comment ils sont crédités. Ils se composent d'artistes, avec (facultativement) leurs noms tels que crédités dans la version spécifique, la piste, etc., et joindre des phrases entre eux. Par exemple, sur la version « Love Sponge » l'artiste est « [Walk off the Earth](#) » mais est crédité comme « Gianni et Sarah ».

Veuillez consulter la page [Artist Credits](#) sur le site Web de MusicBrainz pour plus d'informations.

caa

Le [Cover Art Archive](#) qui est un projet conjoint entre [Internet Archive](#) et [MusicBrainz](#), dont le but est de rendre les images de couvertures accessibles à tous sur Internet de manière organisée et pratique.

Veuillez consulter la page [Cover Art Archive](#) sur le site Web de MusicBrainz pour plus d'informations.

disc id

Un ID du disque est le numéro de code que MusicBrainz utilise pour lier un CD physique à une liste de versions. C'est une chaîne de lettres, comme XzPS7vW.HPHsYemQh0HBUGr8vuU-. Les ID de disque pour une version peuvent être vus dans l'onglet ID de disque pour la version sur MusicBrainz. En cliquant dessus, vous obtiendrez un affichage détaillé de l'ID du disque, y compris la liste des versions jointes.

Une version peut avoir n'importe quel nombre d'identifiants de disque et un identifiant de disque peut être lié à plusieurs versions. Cela est dû au fait que [le calcul de l'ID de disque](#) implique un hachage des décalages d'image des pistes de CD. Différents fabricants de CD ont souvent des décalages de trame légèrement différents, et donc des ID de disque différents.

Inversement, deux CD différents peuvent avoir exactement le même jeu de décalages d'image et donc le même ID de disque. Par exemple, lwh18fGzJyLXQR33ug60E8jhF4k- s'applique à une large [variété de versions](#) par une variété d'artistes.

mbid

L'identifiant MusicBrainz, qui est un code unique utilisé pour identifier chaque élément de la base de données MusicBrainz. Il s'agit d'identifiants universels uniques (UUID) de 128 bits représentés par 32 chiffres hexadécimaux, affichés en cinq groupes séparés par des tirets, sous la forme 8-4-4-4-12 pour un total de 36 caractères.

Veuillez consulter la page [UUID](#) sur Wikipedia pour plus d'informations.

medium

Une des choses physiques et distinctes que vous obtiendriez lorsque vous achetez quelque chose dans un magasin de disques. Ce sont les CD, vinyles, etc. individuels contenus dans l'emballage d'un album (ou de tout autre type de sortie). Les supports sont toujours inclus dans un communiqué, et ont une position dans ledit communiqué (par exemple, disque 1 ou disque 2). Ils ont un format comme CD, vinyle 12 « ou cassette (dans certains cas, cela sera inconnu), et peuvent avoir un titre optionnel (par exemple, disque 2: Les jeunes années). Par exemple, le CD 1 de « [The Wall](#) ».

Veuillez consulter la page [Moyen](#) sur le site Web de MusicBrainz pour plus d'informations.

non-album track

Ce terme est obsolète et a été remplacé par « standalone recording ».

recording

Une entité dans MusicBrainz qui peut être liée à des pistes sur les versions. Chaque piste doit toujours être associée à un seul enregistrement, mais un enregistrement peut être lié à n'importe quel nombre de pistes. Par exemple, cet enregistrement de « [Bohemian Rhapsody](#) » se trouve en tant que piste sur plus de 100 versions.

Veuillez consulter la page [Enregistrement](#) sur le site Web de MusicBrainz pour plus d'informations.

release

Représente la délivrance unique d'un produit à une date spécifique avec des informations de sortie spécifiques telles que le pays, l'étiquette, le code-barres et l'emballage. Par exemple « [Sea of No Cares](#) » est une version de l'album publié par Great Big Sea.

Veuillez consulter la page [Release](#) sur le site Web de MusicBrainz pour plus d'informations.

release group

Regroupe plusieurs versions différentes en une seule entité logique. Chaque version appartient à un et un seul groupe de versions. Les groupes de sortie et les sorties sont des « albums » dans un sens général, mais avec une différence importante: une sortie est quelque chose que vous pouvez

acheter en tant que support tel qu'un CD ou un disque vinyle, tandis qu'un groupe de sortie embrasse le concept global d'un album - peu importe le nombre de CD ou d'éditions / versions dont il disposait. Par exemple, le groupe de versions « [Sea of No Cares](#) » contient plusieurs versions.

Veuillez consulter la page [Release Group](#) sur le site Web de MusicBrainz pour plus d'informations.

standalone recording

Un enregistrement qui n'est lié à aucune sortie. Un exemple est « [Sea of No Cares \(live\)](#) » de Great Big Sea.

Veuillez consulter la page [Enregistrement autonome](#) sur le site Web de MusicBrainz pour plus d'informations.

track

Une piste est la façon dont un enregistrement est représenté sur une version particulière (ou, plus précisément, sur un support particulier). Chaque piste a un titre et est créditée à un ou plusieurs artistes. Par exemple, la piste 7 de l'album « [Back to Boston](#) » de Jason Anderson est « [Driving Home](#) ».

Veuillez consulter la page [Track](#) sur le site Web de MusicBrainz pour plus d'informations.

work

Une création intellectuelle ou artistique distincte, qui peut s'exprimer sous la forme d'un ou plusieurs enregistrements audio. Bien qu'une « œuvre » dans MusicBrainz soit généralement de nature musicale, ce n'est pas nécessairement le cas. Une œuvre peut également être un roman, une pièce de théâtre, un poème ou un essai, enregistré plus tard sous forme d'oratoire ou de livre audio. Par exemple, la chanson « [Blinded by the Light](#) » écrite par Bruce Springsteen a été enregistrée plus de 100 fois.

Veuillez consulter la page [Work](#) sur le site Web de MusicBrainz pour plus d'informations.

Voir aussi:

Pour plus d'informations sur ces termes et sur les autres termes utilisés, veuillez vous reporter à la page [Terminologie](#) sur le site Web MusicBrainz.

Voir aussi:

Pour une explication détaillée de la façon dont tous les éléments sont liés dans l'environnement MusicBrainz, veuillez vous référer à la page Web [MusicBrainz Database / Schema](#).

CHAPTER 5

Commencer

5.1 Téléchargez et installez Picard

MusicBrainz Picard est disponible pour tous les principaux systèmes d'exploitation de bureau (par exemple Windows, Linux et macOS), et sous plusieurs formes (exécutables de version formelle directement téléchargeables, versions de gestionnaire de packages de ceux-ci, exécutables de construction quotidiens, code source Python que vous pouvez exécuter avec le vôtre Environnement Python, etc.)

On s'attend à ce que la plupart des utilisateurs exécutent des exécutables de version formelle ou des équivalents de gestionnaire de paquets car ils sont faciles à installer et sont des versions stables qui sont moins susceptibles d'avoir des bogues dans les fonctionnalités expérimentales ou nouvelles.

Cependant, tout utilisateur souhaitant contribuer au développement de Picard ou de ses plugins peut vouloir exécuter à partir du code source, en le téléchargeant depuis GitHub en utilisant une version de Git sur son propre ordinateur. Si vous souhaitez contribuer au code Picard mais que vous ne comprenez pas ce que disait la phrase précédente, alors vous avez un peu de courbe d'apprentissage. :-)

La dernière version de MusicBrainz Picard est toujours disponible au téléchargement sur [le site Web de Picard](#). Cela inclut les programmes d'installation pour toutes les plates-formes prises en charge ainsi que [le code source de la version](#). Le tout dernier code source est également disponible dans [le référentiel GitHub](#).

5.1.1 Installation de Picard sous Linux

Installation avec Flatpak

Picard est disponible sur [Flathub](#). Cette version devrait fonctionner sur toutes les distributions Linux modernes, à condition que Flatpak soit installé (voir [Installation rapide du Flatpak](#)).

Commencez par activer le référentiel Flathub:

```
flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.  
flatpakrepo
```

Vous pouvez maintenant installer Picard:

```
flatpak install flathub org.musicbrainz.Picard
```

Installation avec Snap

Picard est également disponible sur Snap. Cette version devrait fonctionner sur toutes les distributions Linux modernes, tant que Snap est installé (voir [Installation de Snap](#)).

La page [Snap Store de Picard](#) donne des instructions détaillées sur la façon d'installer Picard sur diverses distributions Linux. Si votre distribution Linux le supporte, vous pouvez installer Picard depuis le centre logiciel de votre distribution, par exemple Ubuntu Software ou KDE Discover. Vous pouvez également installer Picard à partir du terminal:

```
snap install picard
```

Note: Picard installé en tant que Snap s'exécute dans un bac à sable et n'a donc pas un accès complet à tous les fichiers et dossiers de votre système. Par défaut, Picard a accès à votre dossier personnel. Vous pouvez en outre lui donner accès à un support amovible en exécutant la commande suivante sur un terminal:

```
snap connect picard:removable-media
```

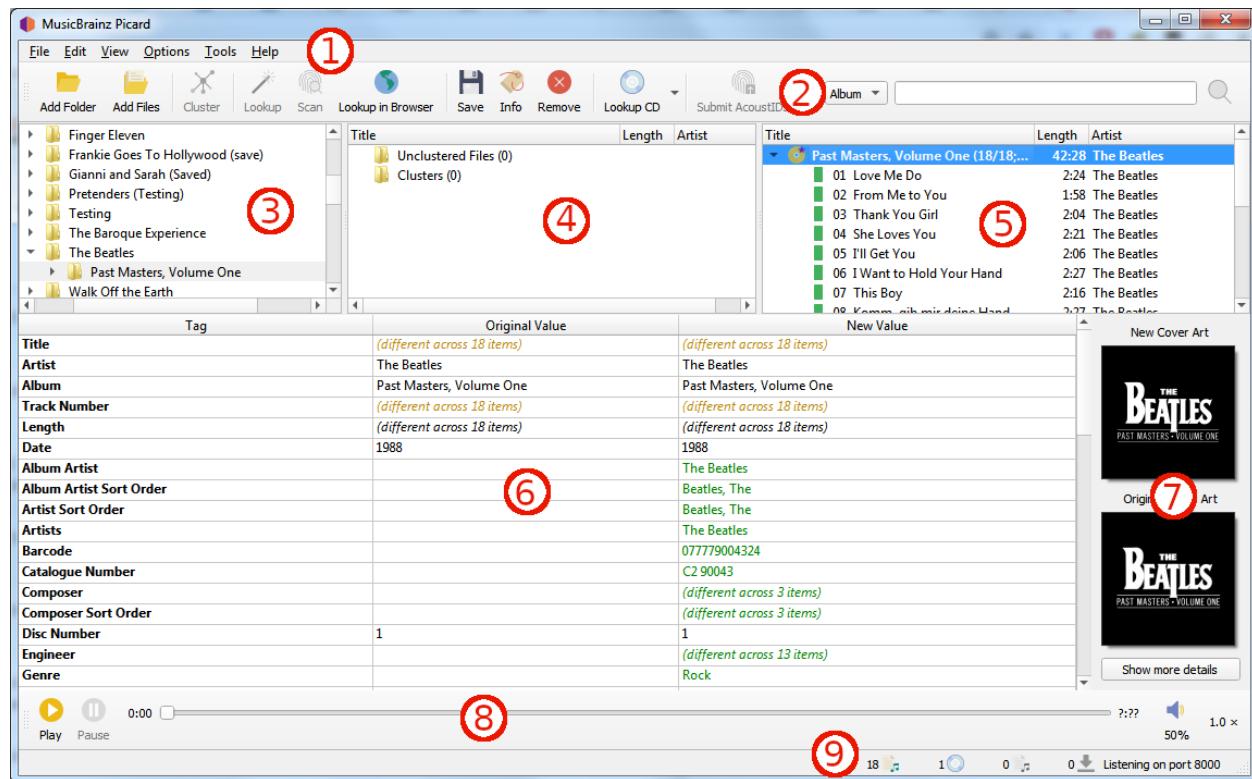
Installation à partir du référentiel de packages de votre distribution

Picard est disponible dans les référentiels de paquets de la plupart des distributions. La [page de téléchargement](#) fournit des liens vers les packages pour les distributions Linux courantes. Veuillez consulter la documentation de votre distribution pour savoir comment installer les packages logiciels.

Veuillez noter que la plupart des distributions contiennent généralement des versions plus anciennes de Picard. Si vous souhaitez utiliser la dernière version disponible,

comme cela est recommandé, installez Picard en tant que Flatpak ou Snap comme décrit ci-dessus.

5.2 Écran principal

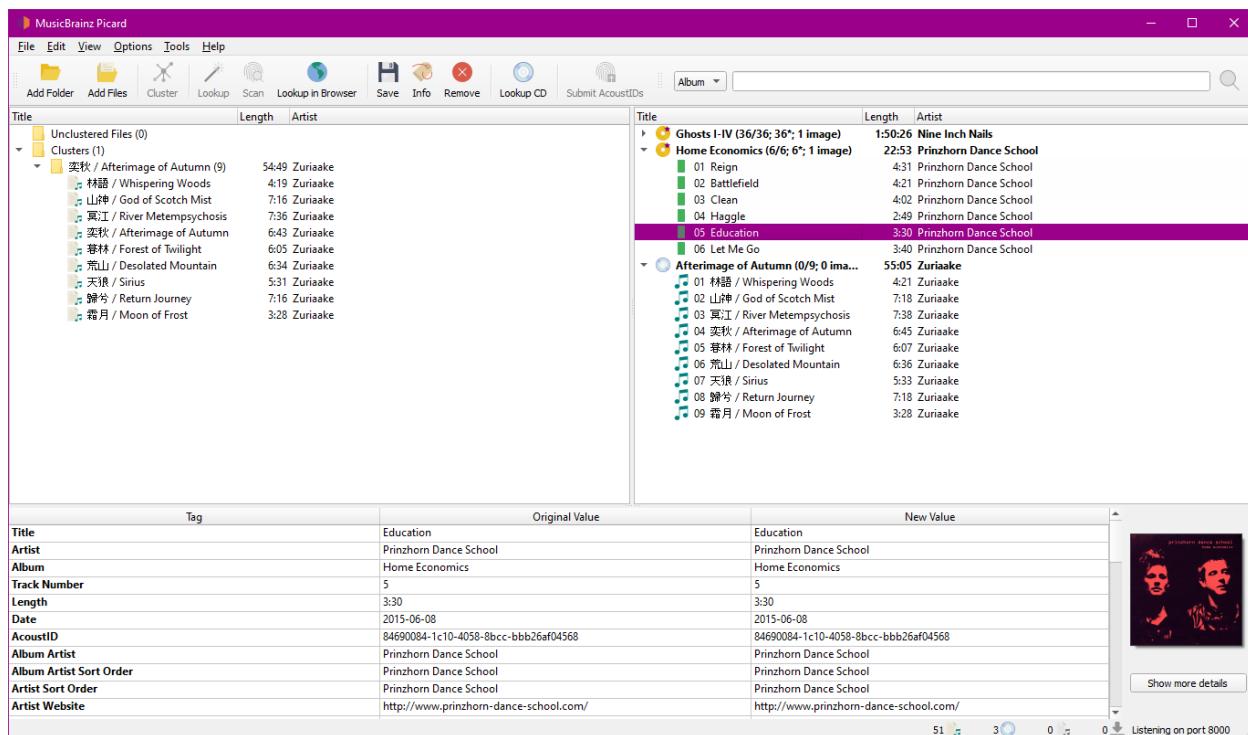


- 1. Barre de menus:** Ceci fournit le menu déroulant des actions que Picard peut effectuer.
- 2. Barre d'outils:** Ceci fournit des liens rapides vers les principales fonctions exécutées par Picard. Ceci peut être personnalisé par l'utilisateur dans les paramètres *User Interface Options*.
- 3. Navigateur de fichiers:** Ceci fournit un navigateur pour sélectionner les fichiers et répertoires à traiter.
- 4. Volet de cluster:** Souvent appelé « volet de gauche », cette section permet à l'utilisateur de sélectionner et de regrouper les fichiers pour l'analyse, la recherche ou la mise en correspondance.
- 5. Volet Album:** Souvent appelé « volet de droite », cette section affiche les albums extraits de MusicBrainz. C'est la section où les fichiers sont mis en correspondance avec les informations de piste téléchargées.
- 6. Volet Métadonnées:** Cette section est un tableau à trois colonnes des métadonnées de balise pour l'album ou la piste actuellement sélectionné dans le volet Album. La première colonne affiche le nom de la balise, la seconde la valeur d'origine trouvée dans le fichier et la troisième colonne affiche la nouvelle valeur qui sera écrite.
- 7. Pochette:** Ceci montre la nouvelle image de la pochette qui sera écrite dans l'album ou la piste sélectionné, ainsi que l'image de la pochette d'origine trouvée

dans les fichiers correspondant à l'album ou à la piste sélectionné.

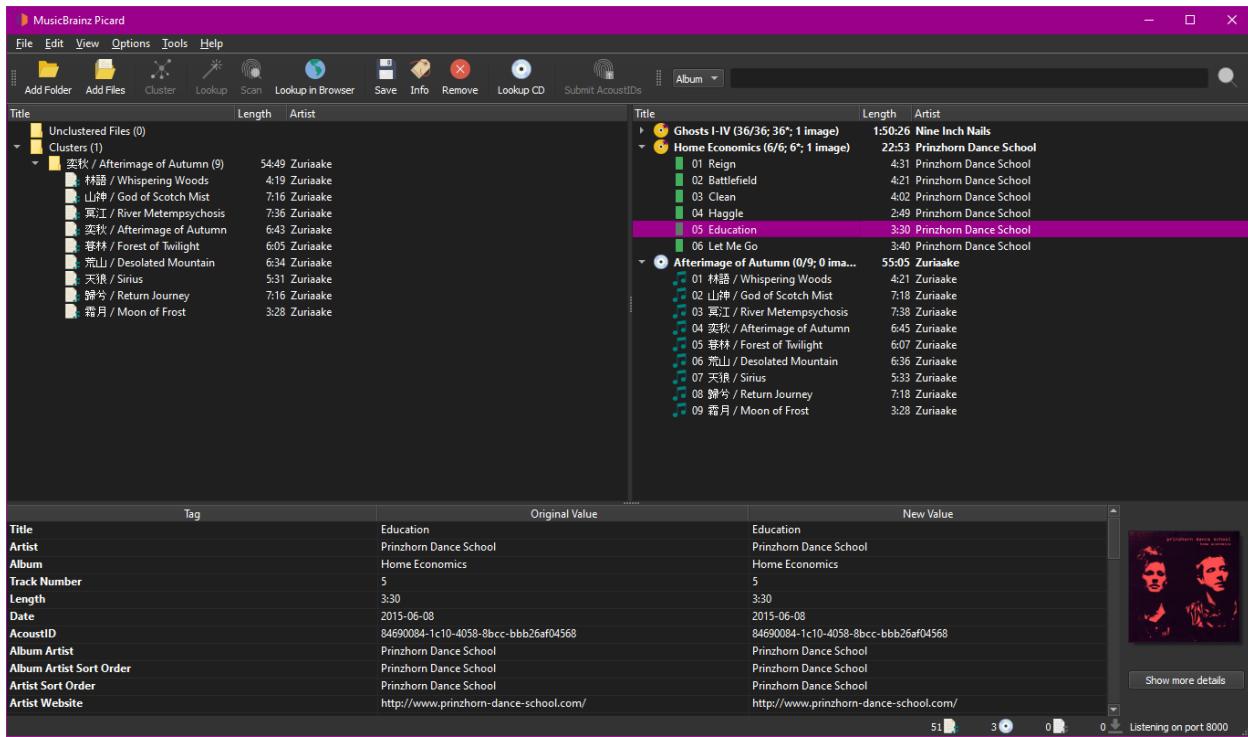
8. **Lecteur de musique:** Le lecteur intégré qui peut être utilisé pour lire des fichiers audio sélectionnés.
9. **Barre d'état:** La barre en bas de l'écran affiche des informations sur le fonctionnement actuel de Picard, y compris des éléments tels que le nombre de fichiers, d'albums et de téléchargements en attente.

Picard est disponible pour différents systèmes d'exploitation. Et bien que les fonctionnalités de Picard soient globalement les mêmes, l'aspect spécifique de l'interface utilisateur peut être légèrement différent en fonction du système d'exploitation. Tout au long de la documentation, des captures d'écran prises sur différents systèmes d'exploitation sont utilisées. Vous trouverez ci-dessous une sélection de captures d'écran de l'écran principal de Picard sur différents systèmes:

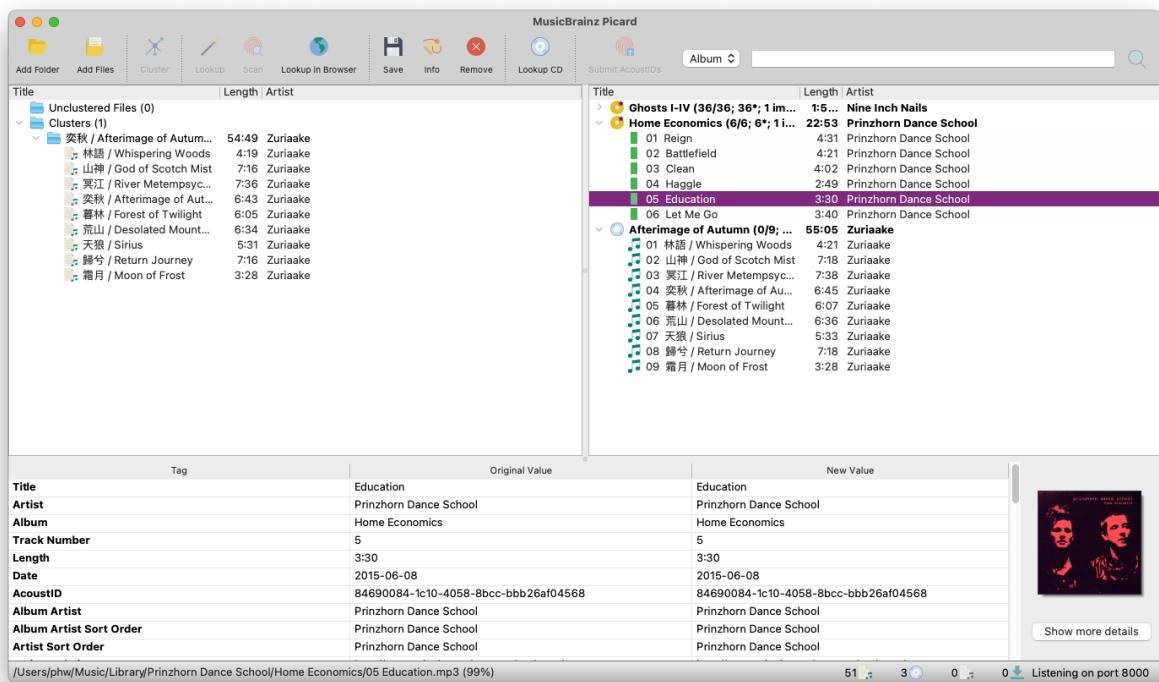


Picard sur Windows 10 (avec interface utilisateur légère)

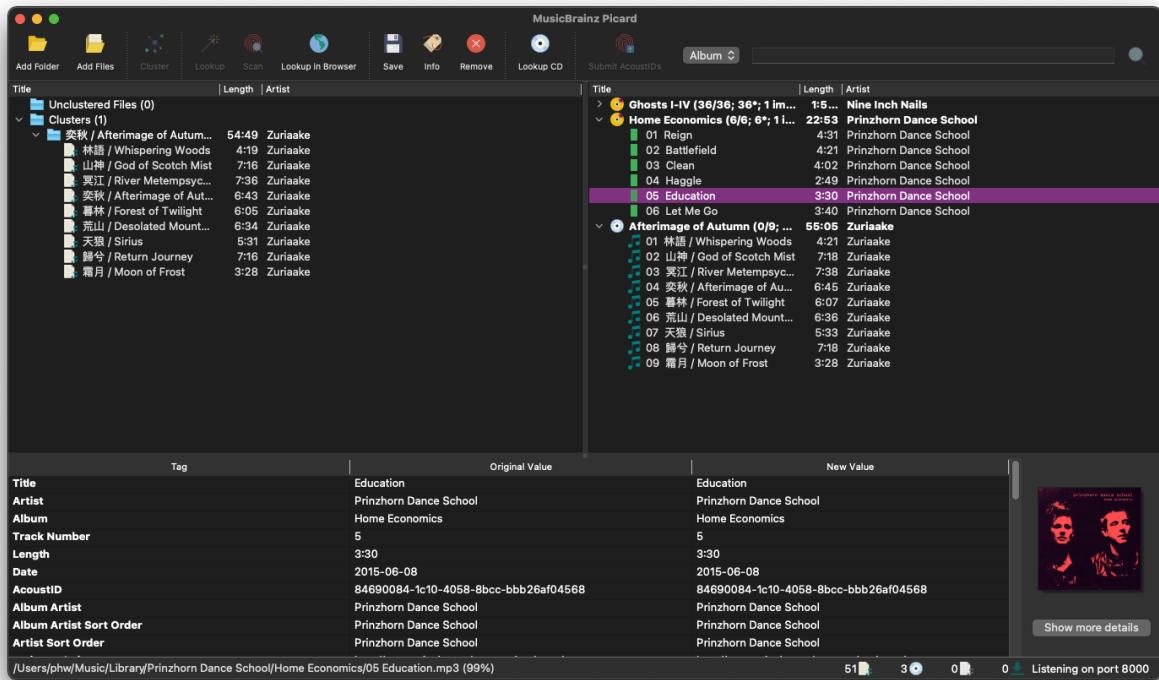
MusicBrainz Picard, Version v2.6.3



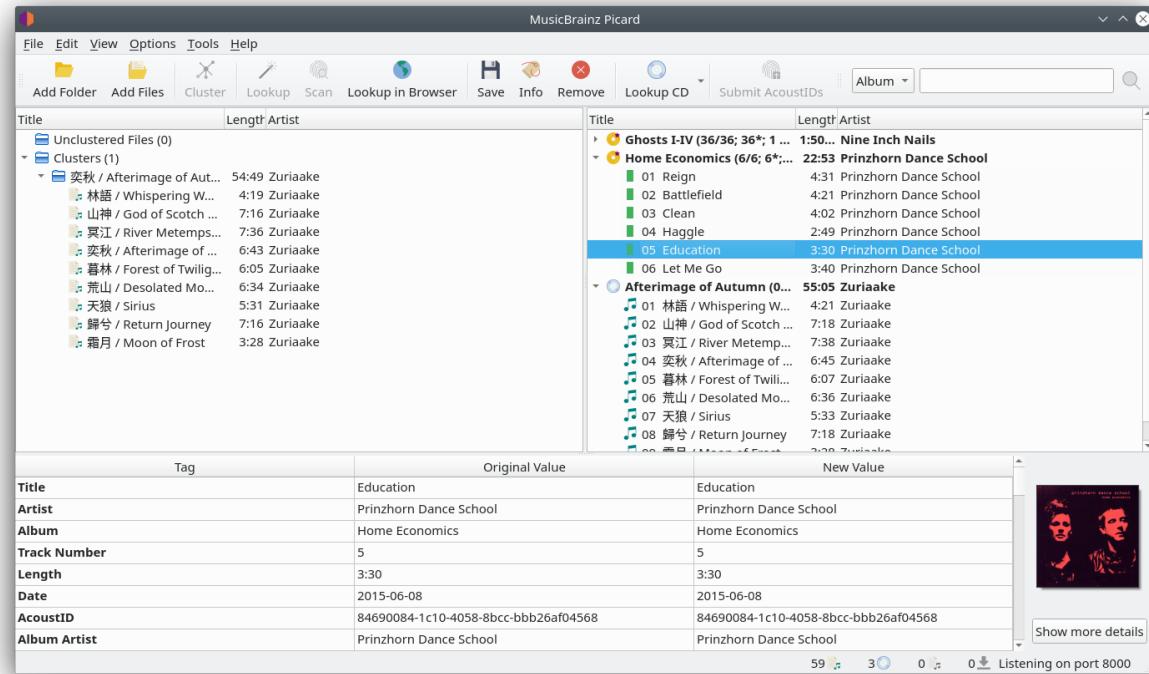
Picard sur Windows 10 (avec une interface utilisateur sombre)



Picard sur macOS



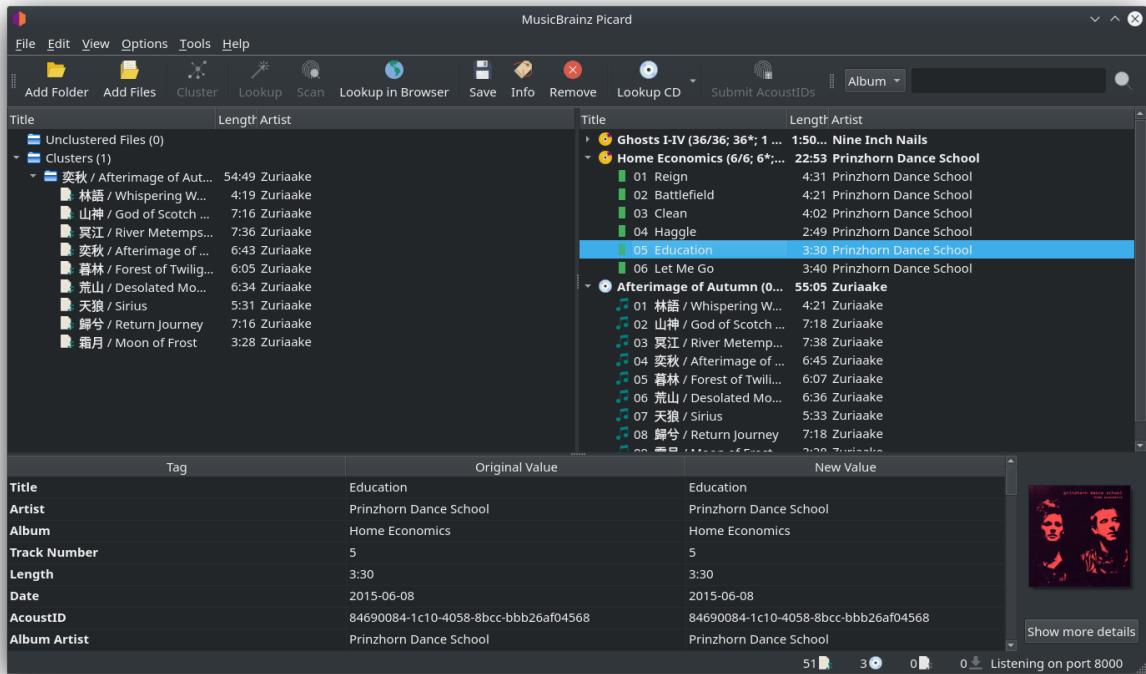
Picard sur macOS (mode sombre)



Picard sous Linux avec l'environnement de bureau KDE Plasma (thème clair)

5.2. Écran principal

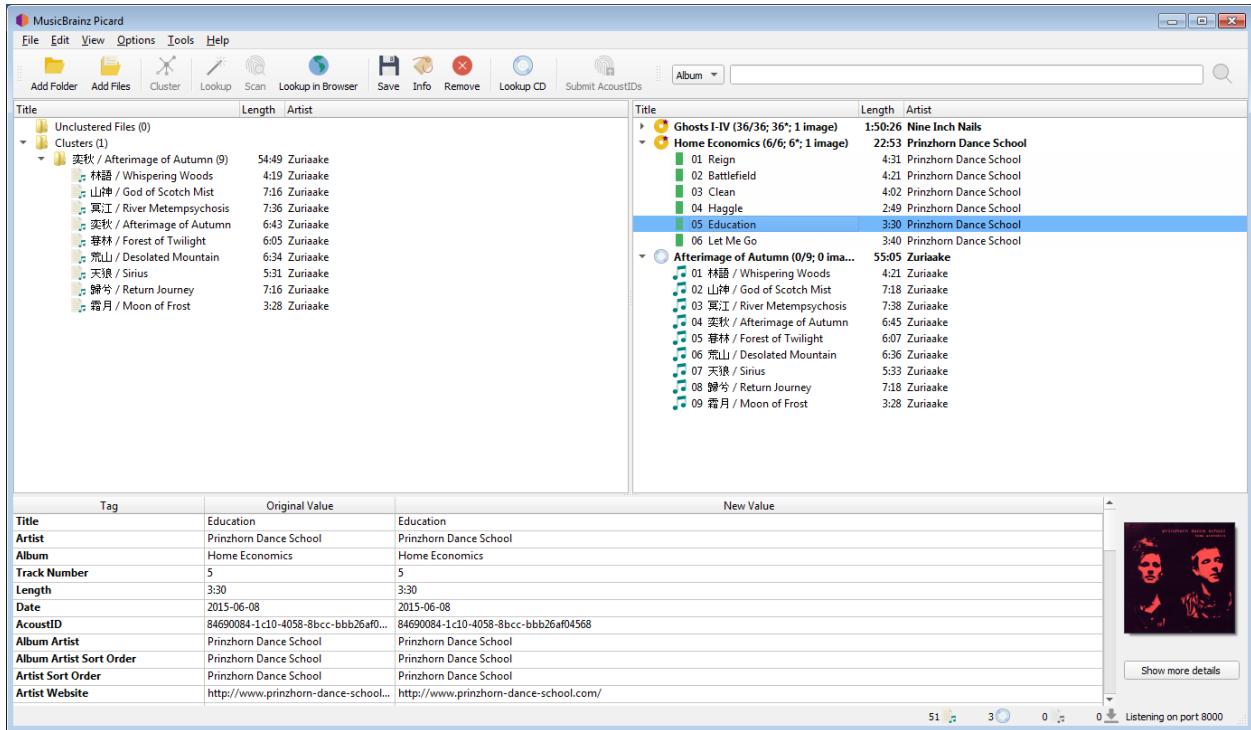
MusicBrainz Picard, Version v2.6.3



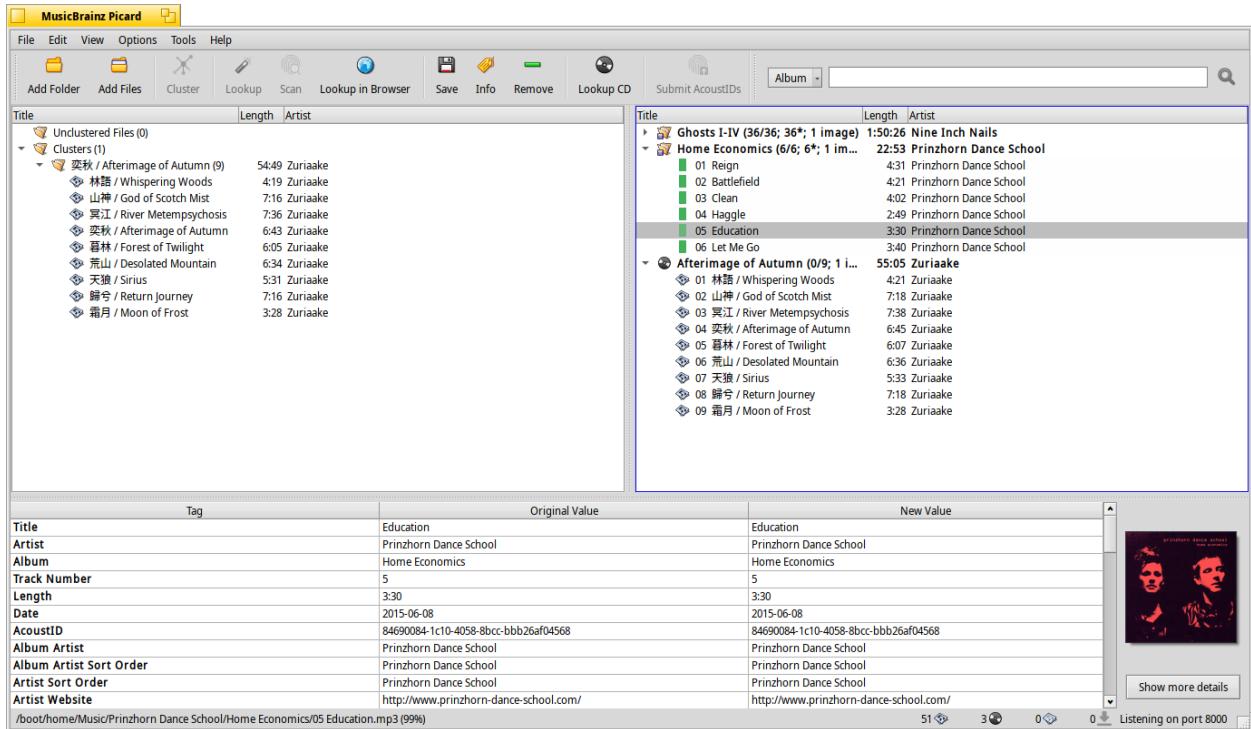
Picard sous Linux avec l'environnement de bureau KDE Plasma (thème sombre)



Picard sous Linux avec l'environnement de bureau GNOME



Picard sur Windows 7



Picard sur Haiku

5.2. Écran principal

5.3 Icônes d'état

Lorsque les albums et les pistes sont affichés dans le volet de droite, chaque ligne commence par une icône pour indiquer l'état de l'élément.

5.3.1 Icônes d'album / de sortie



Cette icône indique que les informations relatives à la version ont été récupérées avec succès dans la base de données MusicBrainz. Certaines pistes, mais pas toutes, peuvent avoir été associées à des fichiers et les informations n'ont pas été modifiées.



Cette icône indique que certaines des pistes ont été mises en correspondance et que les informations de la version ont été modifiées.



Cette icône indique que toutes les pistes ont été mises en correspondance et que les informations n'ont pas été modifiées.



Cette icône indique que toutes les pistes ont été mises en correspondance et que les informations de la version ont été modifiées.



Cette icône indique que Picard a rencontré une erreur avec la version, généralement lors de la récupération des informations de la base de données MusicBrainz.

5.3.2 Icônes de piste



Cette icône indique que la piste est une piste audio et qu'aucun fichier ne correspond actuellement. Cela apparaît s'il n'y a pas de fichier correspondant, ou si plusieurs fichiers correspondent.



Cette icône indique que la piste est une piste vidéo et qu'aucun fichier ne correspond actuellement. Cela apparaît s'il n'y a pas de fichier correspondant, ou si plusieurs fichiers correspondent.



Cette icône indique que la piste est une piste de données et qu'aucun fichier ne correspond actuellement. Cela apparaît s'il n'y a pas de fichier correspondant, ou si plusieurs fichiers correspondent.



Ces icônes indiquent la qualité de la correspondance entre les informations du fichier et les informations de la piste fournies par la base de données MusicBrainz. Le rouge indique une mauvaise correspondance, passant à tout le vert, ce qui indique une très bonne correspondance.



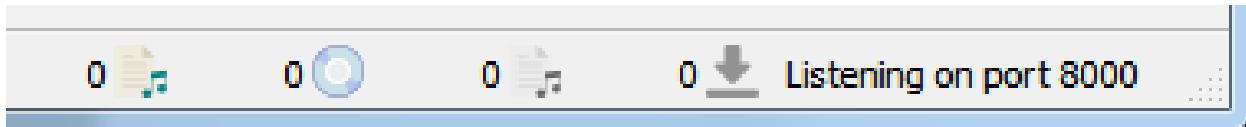
Cette icône indique que la trace a été enregistrée avec succès.



Cette icône indique que Picard a rencontré une erreur en essayant de sauvegarder la trace. Cela est généralement dû au fait que le fichier est marqué en lecture seule ou que vous ne disposez pas des autorisations suffisantes pour enregistrer le fichier dans le répertoire spécifié.

5.3.3 Barre d'état

Il y a une barre d'état en bas de l'écran principal de Picard, qui affiche quelques informations sur l'état actuel du traitement. Cela inclut quatre chiffres ainsi que le numéro du port actuel que Picard surveille.



De gauche à droite, les quatre chiffres représentent:

1. Le nombre de fichiers chargés.
2. Le nombre de versions de MusicBrainz chargées.
3. Le nombre de fichiers en attente d'une action (par exemple, chargement, sauvegarde, prise d'empreinte).
4. Le nombre de demandes de réseau actives.

Vous pouvez savoir si Picard est ou non en train de traiter vos fichiers en regardant les nombres de fichiers en attente et de requêtes réseau en attente. Si les deux sont à 0, rien n'est en attente et le traitement est terminé.

CHAPTER 6

Configuration

Une fois Picard installé sur votre système, l'étape suivante consiste à le configurer selon vos préférences. La configuration consiste à activer les sections d'écran souhaitées pour l'affichage, à sélectionner les actions souhaitées et à définir les différentes options.

6.1 Configuration de l'écran

La configuration de l'écran se trouve sous l'élément « *View* » de la barre de menu. Pour activer l'affichage d'un élément, cochez simplement la case de l'option écran. Les éléments sont:

Navigateur de fichiers

Ceci affiche un navigateur de fichiers sur le côté gauche de l'écran pour sélectionner les fichiers et répertoires à traiter. Les fichiers peuvent être chargés dans Picard en les faisant glisser et en les déposant dans les volets de droite, en double-cliquant sur des fichiers individuels ou en sélectionnant plusieurs fichiers et dossiers et en sélectionnant « Charger les fichiers sélectionnés » dans le menu contextuel.

Les fichiers et répertoires peuvent également être sélectionnés à l'aide du navigateur de fichiers de votre système en les faisant glisser et en les déposant sur l'application Picard.

Couverture Art

Cela affiche la pochette de l'élément actuellement sélectionné (piste ou version) dans une fenêtre à droite de la section des balises de l'affichage.

Cela vous permet de sélectionner ou de remplacer la pochette enregistrée avec la version.

Actions

Cela affiche la barre de boutons des actions effectuées par Picard, située juste en dessous de la barre de menu.

Chercher

Cela affiche la zone de recherche manuelle à droite de la barre de boutons « Actions ».

Joueur

Ceci affiche le lecteur intégré pour la lecture des fichiers audio sélectionnés.

6.2 Options d'action

Les options d'action se trouvent sous l'élément « *Options* » de la barre de menu. Il y a trois actions disponibles que Picard peut effectuer lors de l'enregistrement des fichiers musicaux sélectionnés:

Renommer les fichiers

Picard renommera chaque fichier conformément au script de nommage.

Déplacer des fichiers

Picard déplacera les fichiers vers le répertoire cible conformément au script de nommage.

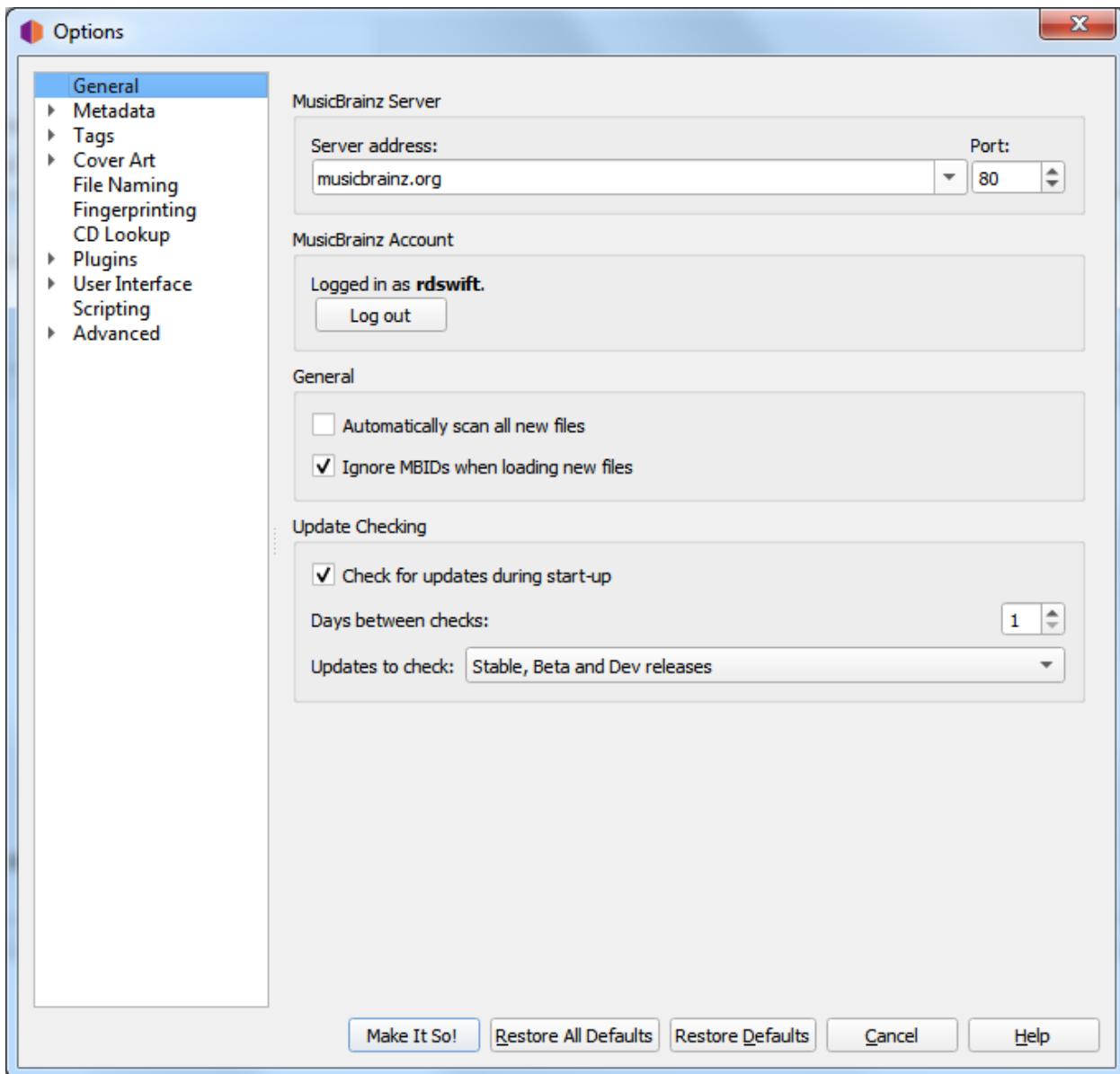
Enregistrer les tags

Picard mettra à jour les balises de métadonnées dans les fichiers conformément aux paramètres d'option et aux scripts de balisage spécifiés.

6.3 Paramètres des options

Les paramètres des options se trouvent sous l'élément « *Options → Options...* » de la barre de menu. Cela ouvrira une nouvelle fenêtre avec les groupes d'options répertoriés dans un format d'arborescence sur le côté gauche et les paramètres individuels sur le côté droit. C'est là que la majorité de la personnalisation de Picard est effectuée.

6.3.1 Options générales



Adresse du serveur

Le nom de domaine du serveur de base de données MusicBrainz utilisé par Picard pour obtenir les détails de votre musique. Valeur par défaut: musicbrainz.org (pour le serveur principal MusicBrainz).

Port

Le numéro de port du serveur. Valeur par défaut: 80 (pour le serveur MusicBrainz principal).

Nom d'utilisateur

Le nom d'utilisateur de votre site Web MusicBrainz, utilisé pour soumettre

des empreintes acoustiques, récupérer et enregistrer des éléments dans vos collections et récupérer des balises personnelles de folksonomy.

Mot de passe

Le mot de passe de votre site Web MusicBrainz.

Analyser automatiquement tous les nouveaux fichiers

Cochez cette case si vous souhaitez que Picard analyse chaque fichier de musique que vous ajoutez et recherche une empreinte digitale AcoustID. Cela prend du temps, mais peut être utile pour vous et MusicBrainz. Laissez-la décochée si vous ne voulez pas que Picard fasse cette analyse automatiquement. Dans tous les cas, vous pouvez demander à Picard de scanner à tout moment un fichier musical particulier en utilisant « *Outils → Analyser* ».

Ignorer les MBID lors du chargement de nouveaux fichiers

Si vous désactivez cette option, Picard n'utilisera pas les identifiants MusicBrainz (MBID) stockés dans les fichiers pour charger automatiquement la version MusicBrainz correspondante et faire correspondre le fichier chargé à la bonne piste. Ceci est utile lors du retraitement de fichiers qui ont déjà été marqués avec des informations incorrectes.

Vérifiez les mises à jour lors du démarrage

Cette option détermine si Picard vérifiera automatiquement les mises à jour du programme au démarrage. Dans tous les cas, vous pouvez demander à Picard de vérifier les mises à jour du programme à tout moment en utilisant « *Aide → Vérifier les mises à jour* ».

Jours entre les chèques

Cette option vous permet de limiter la vérification automatique des mises à jour en sélectionnant l'intervalle, en jours, entre les vérifications. Réglez-le sur 1 si vous souhaitez effectuer une vérification quotidienne, 7 pour les vérifications hebdomadaires, etc. Notez que cela ne s'applique que si l'option « *Vérifier les mises à jour au démarrage* » est activée.

Mises à jour à vérifier

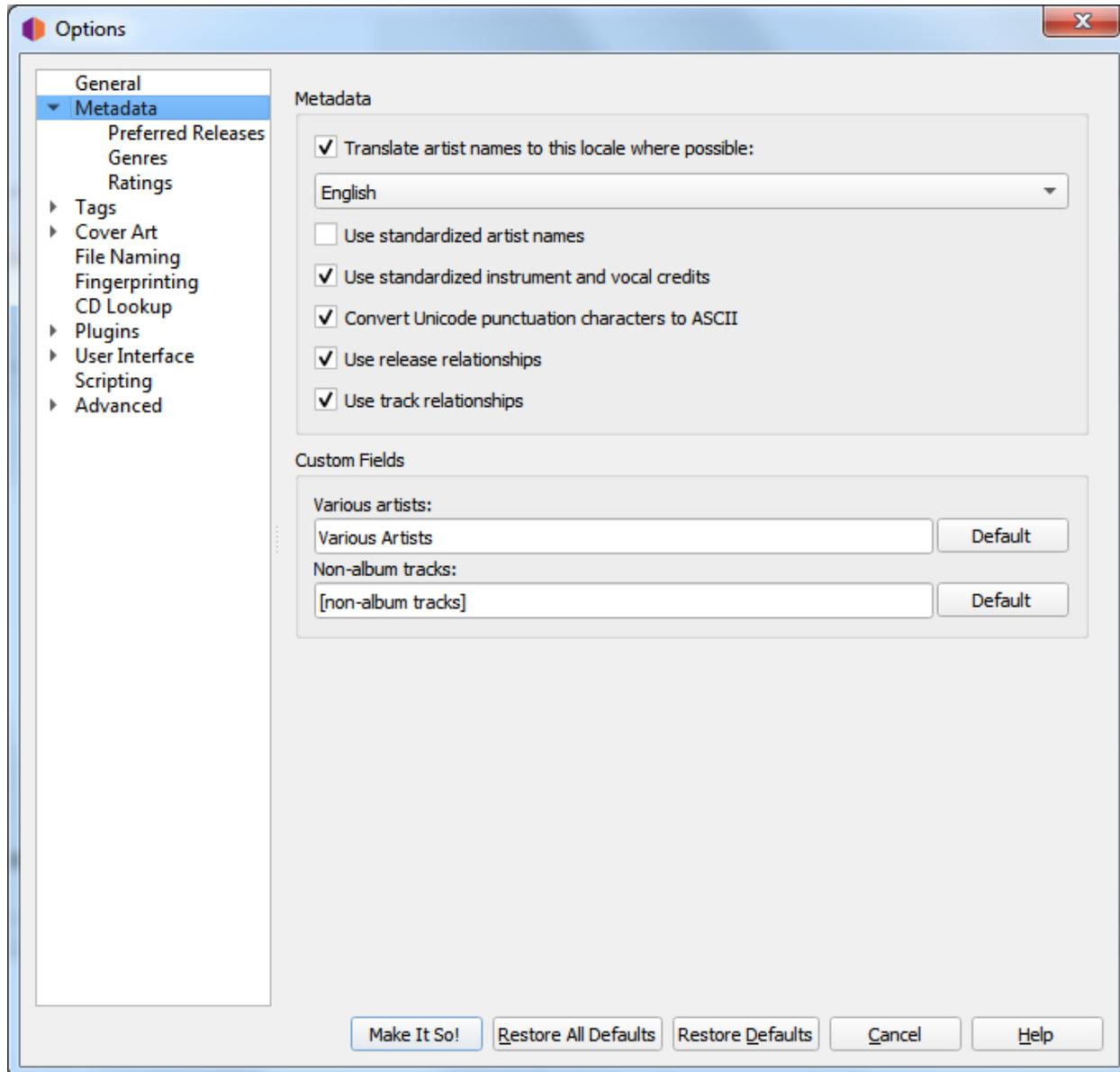
Cette option vous permet de sélectionner les niveaux de mise à jour à vérifier. Vos options sont:

- Versions stables uniquement
- Versions stables et bêta
- Versions stables, bêta et dev

Par exemple, si vous vous abonner à « *Versions stables uniquement* », vous ne serez pas averti si une nouvelle version bêta ou de développement est publiée.

Note: La mise à jour vérifiant les paramètres associés et la commande « Aide → Vérifier les mises à jour... » peuvent ne pas être disponibles lorsque Picard est distribué sous forme de package. Dans ce cas, l'utilisateur doit vérifier auprès du responsable du package pour déterminer quand une mise à jour est disponible.

6.3.2 Options de métadonnées



Traduisez les noms d'artistes dans cette langue lorsque cela est possible

Lorsqu'elle est cochée, Picard verra si un artiste a un alias pour la langue sélectionnée. Si c'est le cas, Picard utilisera cet alias au lieu du nom de

l'artiste lors du marquage. Lorsque « English » est la langue sélectionnée, le nom de tri de l'artiste (qui est, par Style Guideline, stocké en script latin) est utilisé comme solution de secours s'il n'y a pas d'alias anglais.

Utilisez des noms d'artistes standardisés

Cochez pour n'utiliser que des noms d'artistes standard, plutôt que des crédits d'artiste qui peuvent différer légèrement selon les pistes et les versions.

Note: Si l'option « Traduire les noms d'artistes » ci-dessus est également cochée, elle remplacera cette option si un alias approprié est trouvé.

Utilisez des crédits d'instruments et vocaux standardisés

Cochez pour n'utiliser que des noms standard pour les instruments et les voix dans les relations interprètes. Décochez pour utiliser les instruments et les voix comme crédités dans la relation.

Convertir les caractères de ponctuation Unicode en ASCII

Convertit les caractères de ponctuation Unicode des données MusicBrainz en ASCII pour une utilisation cohérente de la ponctuation dans les balises. Par exemple, les guillemets simples droits sont convertis en apostrophes ASCII (") et les ellipses horizontales sont converties en trois points (...).

Utiliser les relations de version

Cochez pour récupérer et écrire les relations au niveau de la version (par exemple: URL, compositeur, parolier, interprète, chef d'orchestre ou mixeur DJ) dans vos fichiers. Vous devez l'activer pour utiliser Picard pour récupérer l'art de couverture.

Utiliser les relations de suivi

Cochez pour écrire des relations au niveau de la piste (par exemple: compositeur, parolier, interprète ou remixeur) dans vos fichiers.

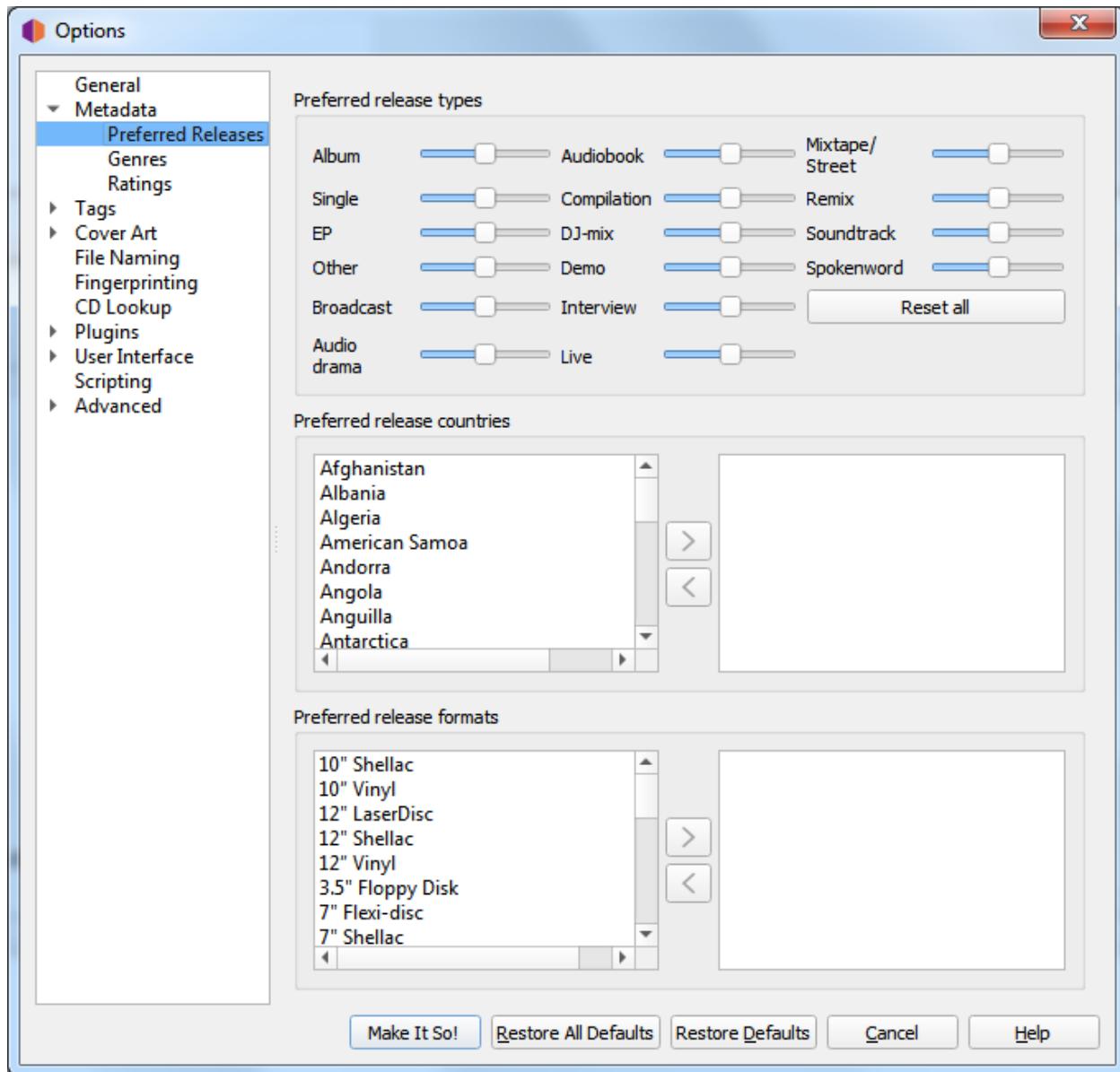
Artistes variés

Choisissez comment vous voulez que l'artiste « Various Artists » soit orthographié.

Pistes hors album

Choisissez la manière dont vous souhaitez regrouper les « pistes hors album ».

Versions préférées



Types de version préférés

Ajustez les curseurs pour différents types de versions afin d'ajuster la probabilité que Picard fasse correspondre un fichier ou un cluster à des versions de différents types. Par exemple, vous pouvez l'utiliser pour réduire la probabilité que Picard fasse correspondre un fichier ou un album à une version Compilation ou Live.

Pays de version préférés

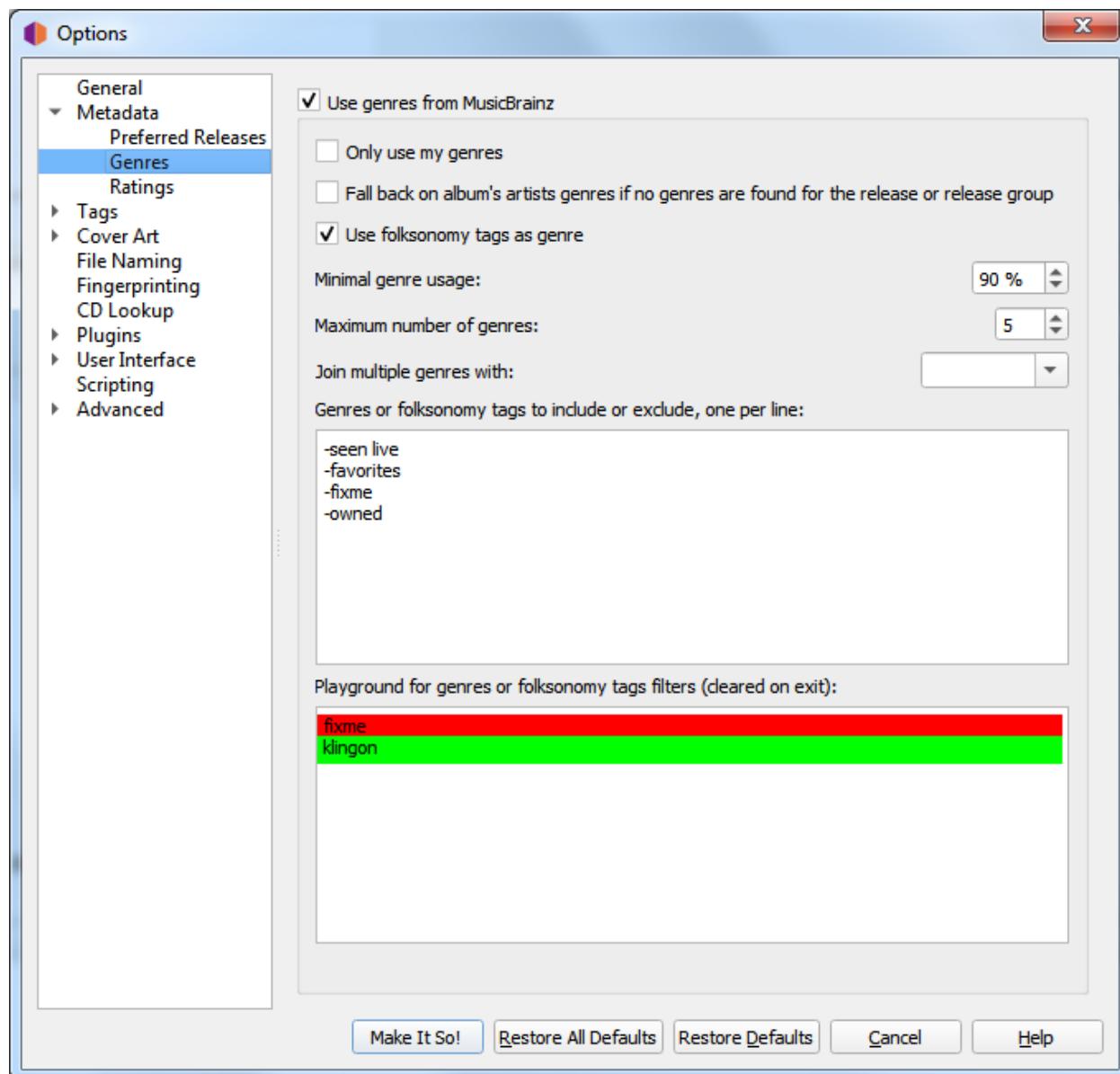
Ajoutez un ou plusieurs pays à la liste pour que Picard préfère les clusters ou fichiers correspondants aux versions des pays choisis. Cette liste est également utilisée pour hiérarchiser les fichiers dans le menu contextuel

« Autres versions ».

Formats de version préférés

Ajoutez un ou plusieurs formats à la liste pour que Picard préfère les clusters ou fichiers correspondants aux versions du format spécifié. Cette liste est également utilisée pour hiérarchiser les fichiers dans le menu contextuel « Autres versions ».

Genres



Utilisez les genres de MusicBrainz

Utilisez les genres fournis par MusicBrainz et enregistrez-les dans la balise de genre.

Revenez aux genres d'artistes de l'album si aucun genre n'est trouvé pour la sortie ou le groupe de sortie

S'il n'y a pas de genre défini pour la sortie ou le groupe de sortie sur MusicBrainz, utilisez plutôt le genre de l'artiste de l'album.

N'utiliser que mes genres

Lorsqu'il est activé, Picard n'écrira que les genres que vous avez personnellement soumis à MusicBrainz. Vous devrez définir votre nom d'utilisateur et votre mot de passe pour utiliser cette fonctionnalité.

Utilisez des balises folksonomy comme genres

Cochez pour utiliser toutes les balises folksonomy pour définir le genre. Sinon, seules les balises considérées par MusicBrainz comme des genres appropriés seront utilisées.

Utilisation minimale du genre

Choisissez la popularité du genre avant qu'il ne soit écrit par Picard. Par défaut: 90%. Réduire la valeur ici conduira à plus de genres, mais peut-être moins pertinents, dans vos fichiers.

Nombre maximum de genres

Choisissez le nombre de genres à utiliser. Par défaut: 5. Si vous ne voulez qu'un seul genre, réglez-le sur 1.

Rejoignez plusieurs genres avec

Sélectionnez le caractère à utiliser pour séparer plusieurs genres.

Balises de genres ou de folksonomy à inclure ou à exclure

Une expression par ligne, insensible à la casse. Vous pouvez utiliser le champ de texte « Cour de récréation » pour saisir certains genres et tester les règles que vous avez configurées. Les genres qui seront exclus seront marqués en rouge, les genres inclus seront marqués en vert.

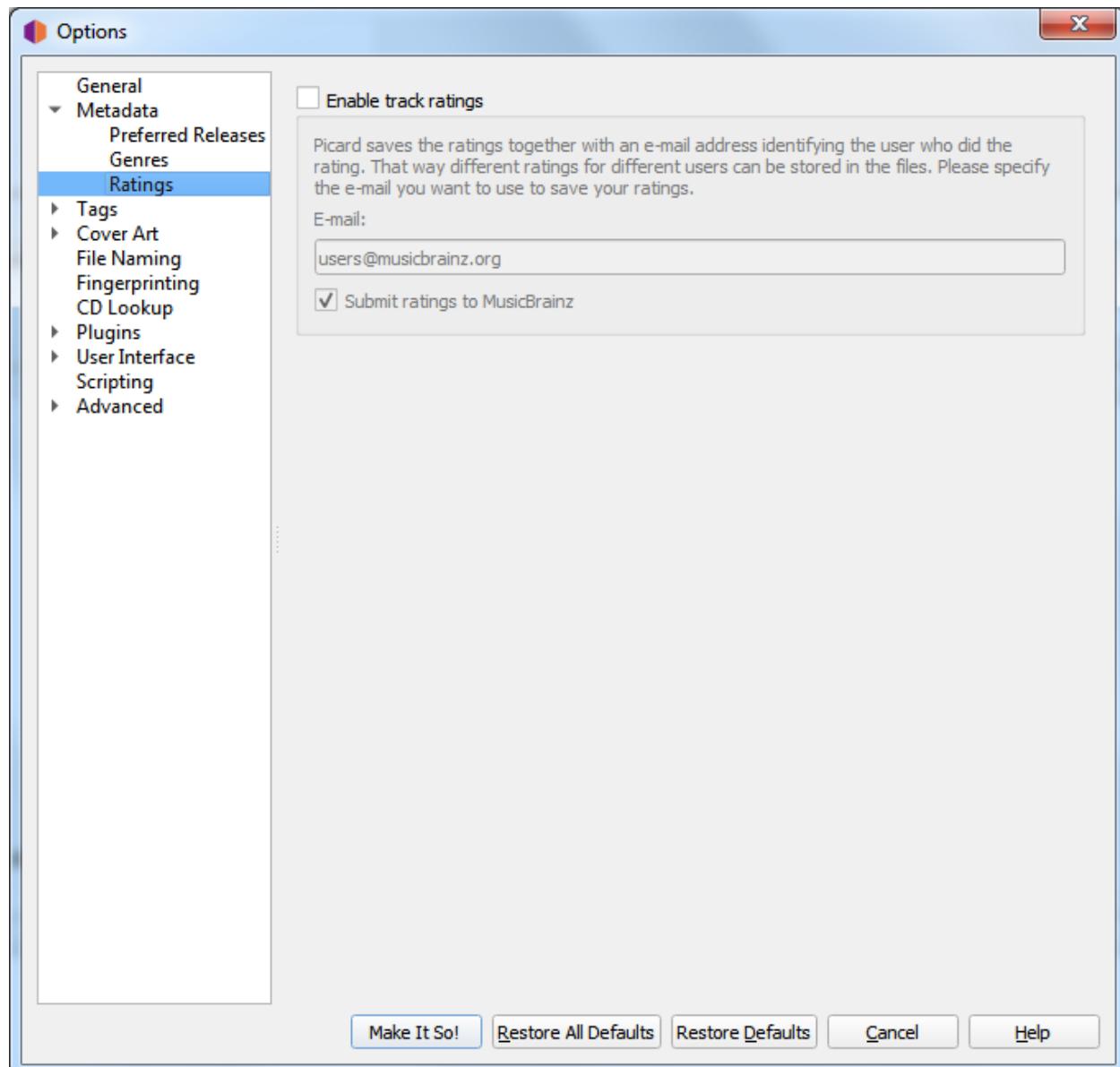
- **Commentaires:** Les lignes ne commençant pas par “-” ou “+” sont ignorées. (par exemple: # commentaire, ! comment ou comment)
- **Filtrage strict:** excluez le mot exact en le préfixant avec “-” (par exemple: -word). Incluez le mot exact, même si une autre règle l'exclurait, en le préfixant avec “+” (par exemple: +mot).
- **Filtrage des caractères génériques:** exclut tous les genres se terminant par « mot » (par exemple: -*mot). Incluez tous les genres commençant par « mot » (par exemple: +mot*). Excluez tous les genres commençant par “m” et se terminant par « t » (par exemple: -m*t).
- **Filtrage des expressions régulières (syntaxe Python « re »):** exclut les genres commençant par “w” suivi de n’importe quel caractère, puis “r” suivi d’au moins un “d” (par exemple: -/^w.rd+/).

Aire de jeu pour les filtres de balises de genres ou de folksonomy:

Cette zone vous permet de saisir des balises de genre, une par ligne, pour tester vos filtres. Si une étiquette est marquée en rouge, elle sera filtrée. Un tag marqué en vert sera autorisé.

Note: Cette liste de balises de test sera effacée lorsque vous quitterez la section de configuration.

Évaluations



Activer les évaluations de piste

Cochez pour écrire les notes de piste dans vos fichiers.

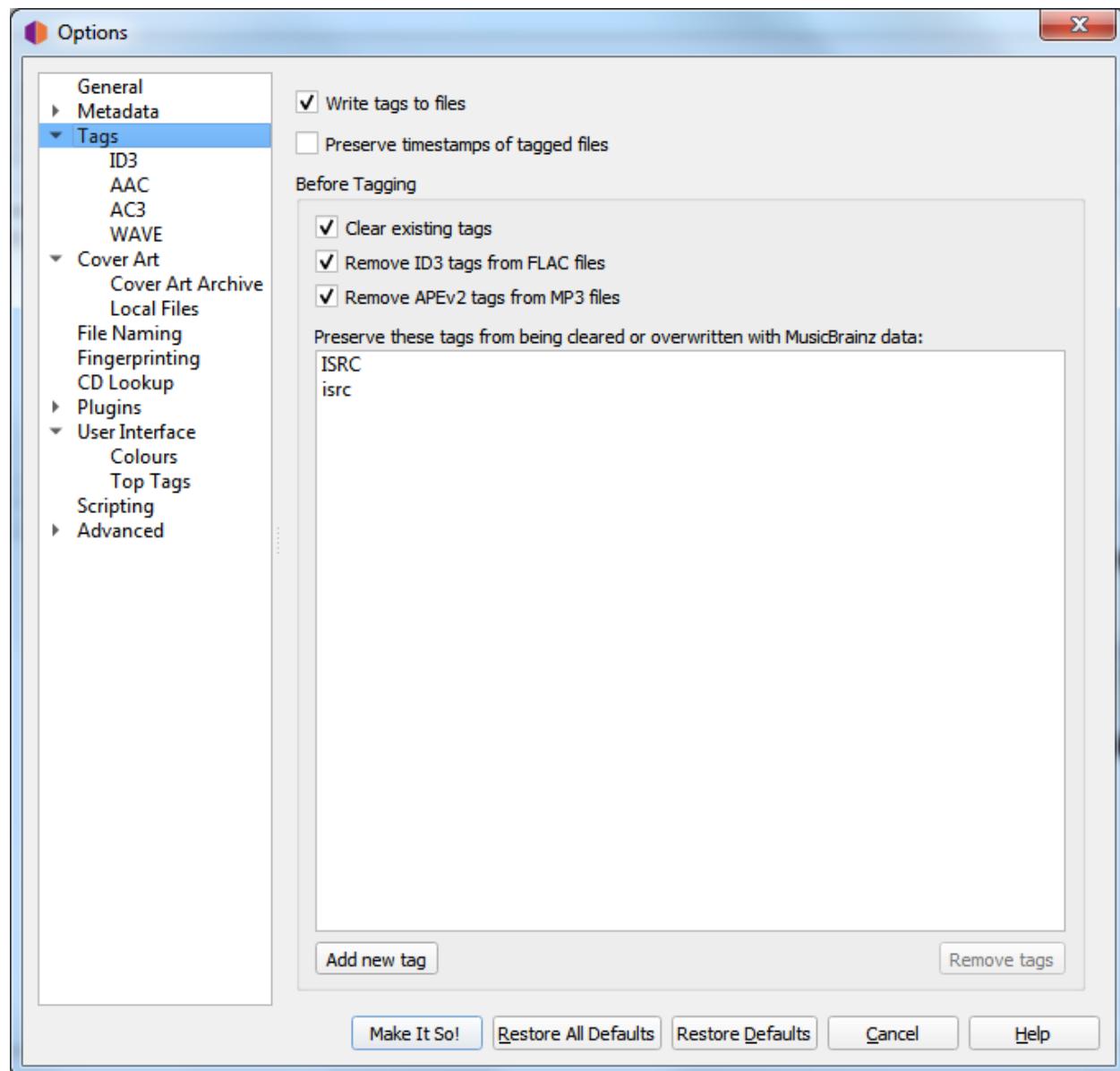
Email

L'adresse e-mail utilisée lors de la soumission des notes à MusicBrainz. Cela identifie l'utilisateur qui a fourni la note.

Soumettez vos notes à MusicBrainz

Cochez pour soumettre des évaluations à MusicBrainz. Les pistes seront notées avec votre compte.

6.3.3 Options de balise



Écrire des balises dans des fichiers

Décochez cette option pour empêcher Picard d'écrire des métadonnées dans

vos fichiers. Picard peut toujours déplacer ou renommer vos fichiers en fonction de vos paramètres.

Conserver les horodatages des fichiers balisés

Si cette case est cochée, Picard ne mettra pas à jour la date et l'heure de la « dernière modification » de vos fichiers musicaux lorsqu'il y écrit de nouvelles balises.

Avant de baliser

Effacer les balises existantes

Cochez cette case pour supprimer toutes les métadonnées existantes et laisser vos fichiers avec uniquement les métadonnées MusicBrainz. Les informations que vous avez peut-être ajoutées via un autre lecteur multimédia, telles que « genre », « commentaires » ou « évaluations », seront supprimées.

Supprimer les balises ID3 des fichiers FLAC

Cochez pour supprimer les balises ID3 des fichiers FLAC - Les commentaires Vorbis sont recommandés pour les fichiers FLAC. Picard écrira les commentaires Vorbis dans les fichiers FLAC quel que soit ce paramètre.

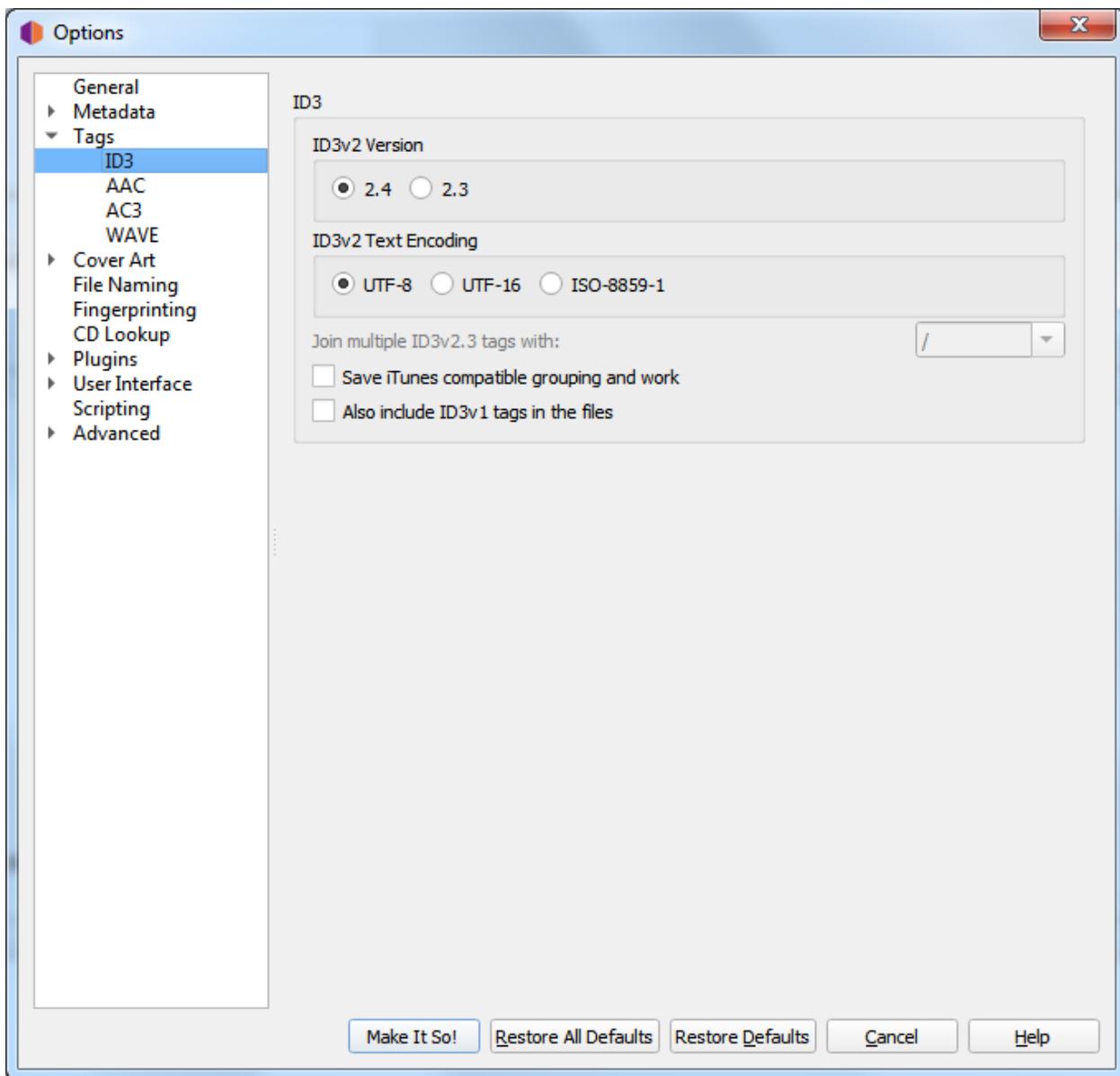
Supprimer les balises APEv2 des fichiers MP3

Cochez pour supprimer les balises APEv2 des fichiers MP3 - ID3 est recommandé pour les fichiers MP3. Picard écrira les balises ID3 sur les fichiers MP3 quel que soit ce paramètre.

Empêchez ces balises d'être effacées ou écrasées par les données MusicBrainz

Ceci est une option avancée: si vous avez des balises que vous devez conserver, entrez leurs noms ici pour empêcher Picard de les écraser.

Fichiers ID3



Version ID3v2

Bien que ID3v2.4 soit la dernière version, son support dans les lecteurs de musique fait toujours défaut. Alors que certains logiciels n'ont aucun problème à utiliser les balises de la version 2.4, d'autres peuvent ne pas être en mesure de lire les balises et d'afficher les informations. La prise en charge d'ID3v2.4 dans d'autres lecteurs multimédias (tels que les smartphones) est variable.

Outre la prise en charge native des balises à valeurs multiples dans la v2.4, la section [Picard Tag Mapping](#) vous montrera ce que vous perdez en choisissant v2.3 au lieu de v2.4.

Encodage de texte ID3v2

La valeur par défaut pour la version 2.4 est UTF-8, la valeur par défaut pour la version 2.3 est UTF-16. N'utilisez ISO-8859-1 que si vous rencontrez des problèmes de compatibilité avec votre lecteur.

Rejoignez les tags ID3v2.3 avec

Comme mentionné ci-dessus, ID3v2.3 ne prend pas en charge les balises à valeurs multiples, et donc Picard les aplatis en chaînes avant de les enregistrer dans les balises ID3v2.3. Ce paramètre définit la chaîne utilisée pour séparer les valeurs lors de l'aplatissement. Utilisation ";" pour la plus grande compatibilité (plutôt que "/" puisque les balises contiennent plus souvent un "/" qu'un ";") et pour la meilleure compatibilité visuelle en Picard entre ID3v2.3 et d'autres formats de balisage.

Note: Ce paramètre est explicitement réservé aux balises ID3 v2.3, car celles-ci ne prennent pas en charge les balises à valeurs multiples appropriées. Par conséquent, ce paramètre de séparateur est en fait plus une solution de contournement à ce fait.

Étant donné que les balises Vorbis permettent à la même balise d'exister plusieurs fois, et donc prennent en charge les balises à valeurs multiples par défaut, il n'y a en fait aucun séparateur. Ce ne sont que des valeurs multiples, affichées en utilisant tout ce que le logiciel du lecteur utilise comme séparateur.

Si, pour une raison quelconque, vous voulez toutes les valeurs de la balise dans un seul champ séparé par un séparateur, vous pouvez utiliser un script comme:

```
$set(musicbrainz_artistid,$join(%musicbrainz_artistid%, / ))
```

pour aplatisir toutes les valeurs dans un seul champ séparé par » / « .

Enregistrez le regroupement compatible iTunes et travaillez

Enregistrez le regroupement des balises et travaillez pour qu'elles soient compatibles avec les versions actuelles d'iTunes. Sans cette option, le regroupement sera affiché dans iTunes en tant que « nom de travail » et le travail ne sera pas disponible.

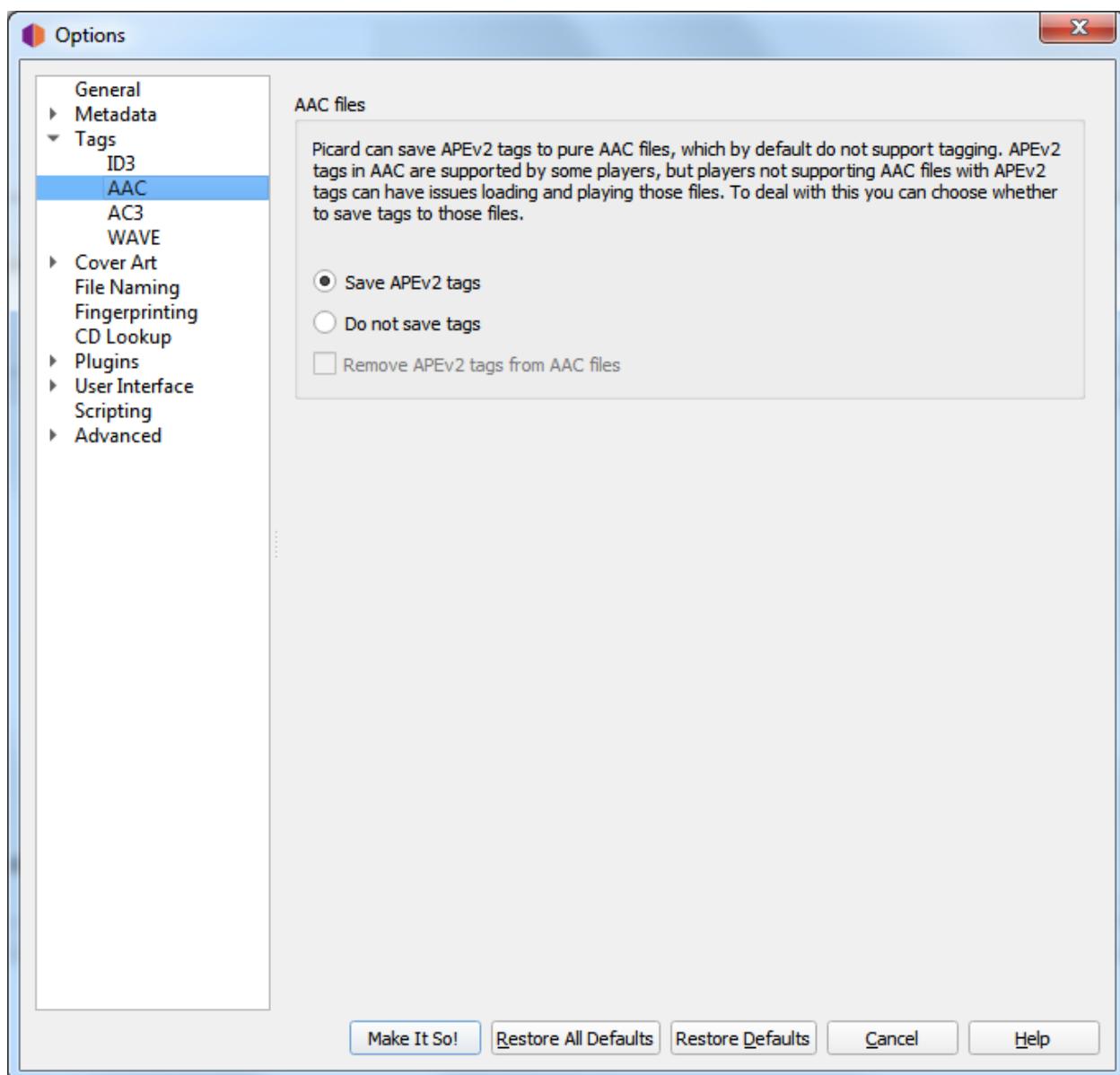
Voir la section *Picard Tag Mapping* pour plus de détails.

Note: Pour les autres joueurs prenant en charge le regroupement et le travail, vous devrez peut-être désactiver cette option. [MusicBee](#) en est un exemple.

** Incluez également les balises ID3v1 dans les fichiers **

Ce n'est pas du tout recommandé. Les balises ID3v1.1 sont obsolètes et peuvent ne pas fonctionner avec des scripts non latins.

Fichiers AAC



Picard peut enregistrer les balises APEv2 dans des fichiers AAC purs, qui par défaut ne prennent pas en charge le balisage. Les balises APEv2 dans AAC sont prises en charge par certains lecteurs, mais les lecteurs qui ne prennent pas en charge les fichiers AAC avec des balises APEv2 peuvent rencontrer des problèmes lors du chargement et de la lecture de ces fichiers. Pour gérer cela, vous pouvez choisir d'enregistrer les balises dans ces fichiers:

** Enregistrer les tags APEv2 **

Picard enregistrera les balises APEv2 dans les fichiers.

Ne pas enregistrer les tags

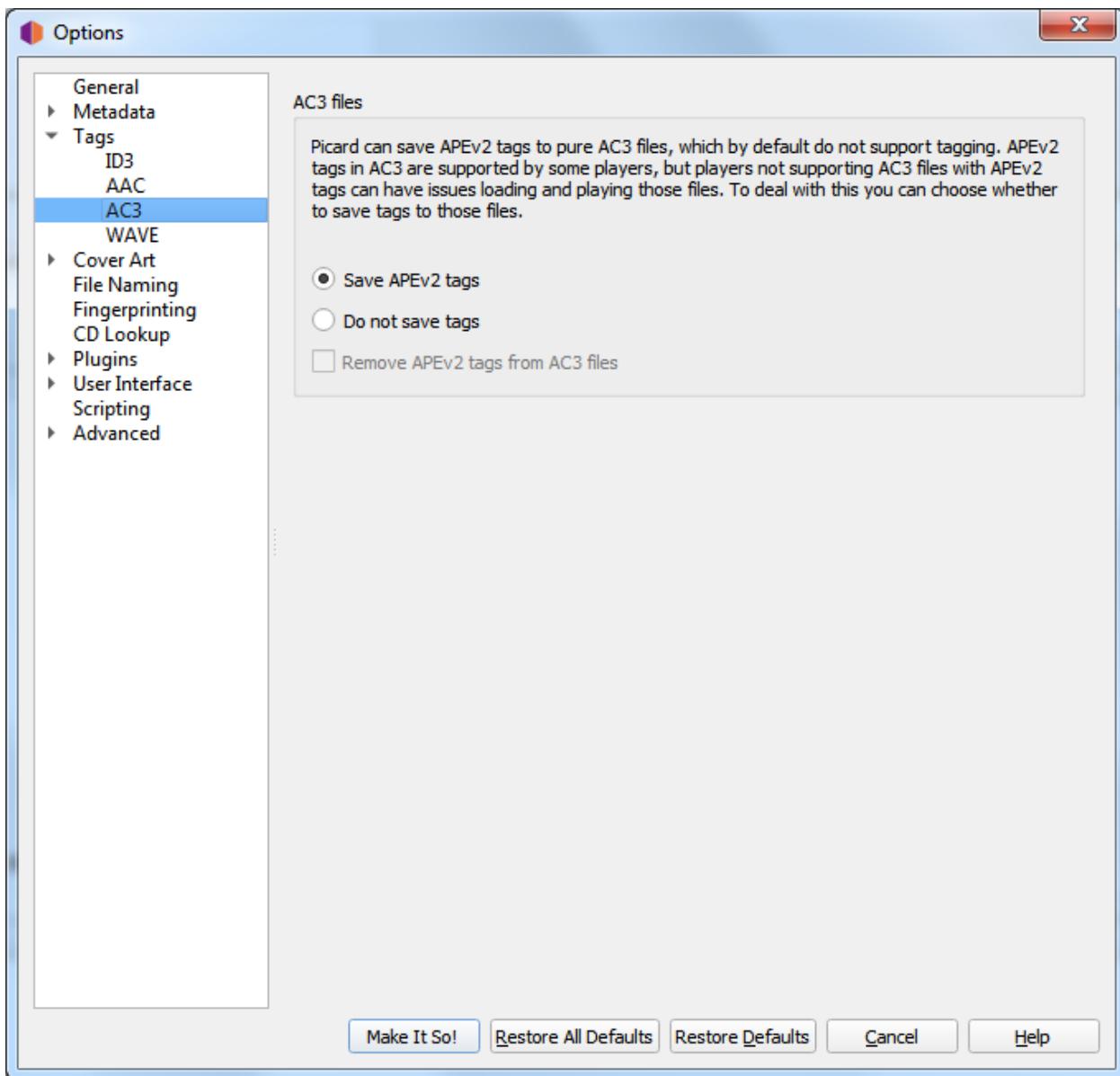
Picard n'enregistrera aucune balise dans les fichiers, mais vous pouvez toujours utiliser Picard pour les renommer. Par défaut, les balises APEv2 existantes seront conservées dans le fichier.

**** Supprimer les balises APEv2 ****

Si vous avez activé « Ne pas enregistrer les balises », cocher cette option obligera Picard à supprimer les balises APEv2 existantes du fichier lors de l'enregistrement.

Quelle que soit la façon dont vous avez configuré l'enregistrement des balises, Picard lira toujours les balises APEv2 existantes dans les fichiers AAC.

Fichiers AC3



Picard peut enregistrer les balises APEv2 dans des fichiers AC3 purs, qui par défaut ne prennent pas en charge le balisage. Les balises APEv2 dans AC3 sont prises en charge par certains lecteurs, mais les lecteurs qui ne prennent pas en charge les fichiers AC3 avec des balises APEv2 peuvent avoir des problèmes pour charger et lire ces fichiers. Pour gérer cela, vous pouvez choisir d'enregistrer les balises dans ces fichiers:

**** Enregistrer les tags APEv2 ****

Picard enregistrera les balises APEv2 dans les fichiers.

Ne pas enregistrer les tags

Picard n'enregistrera aucune balise dans les fichiers, mais vous pouvez toujours utiliser Picard pour les renommer. Par défaut, les balises APEv2

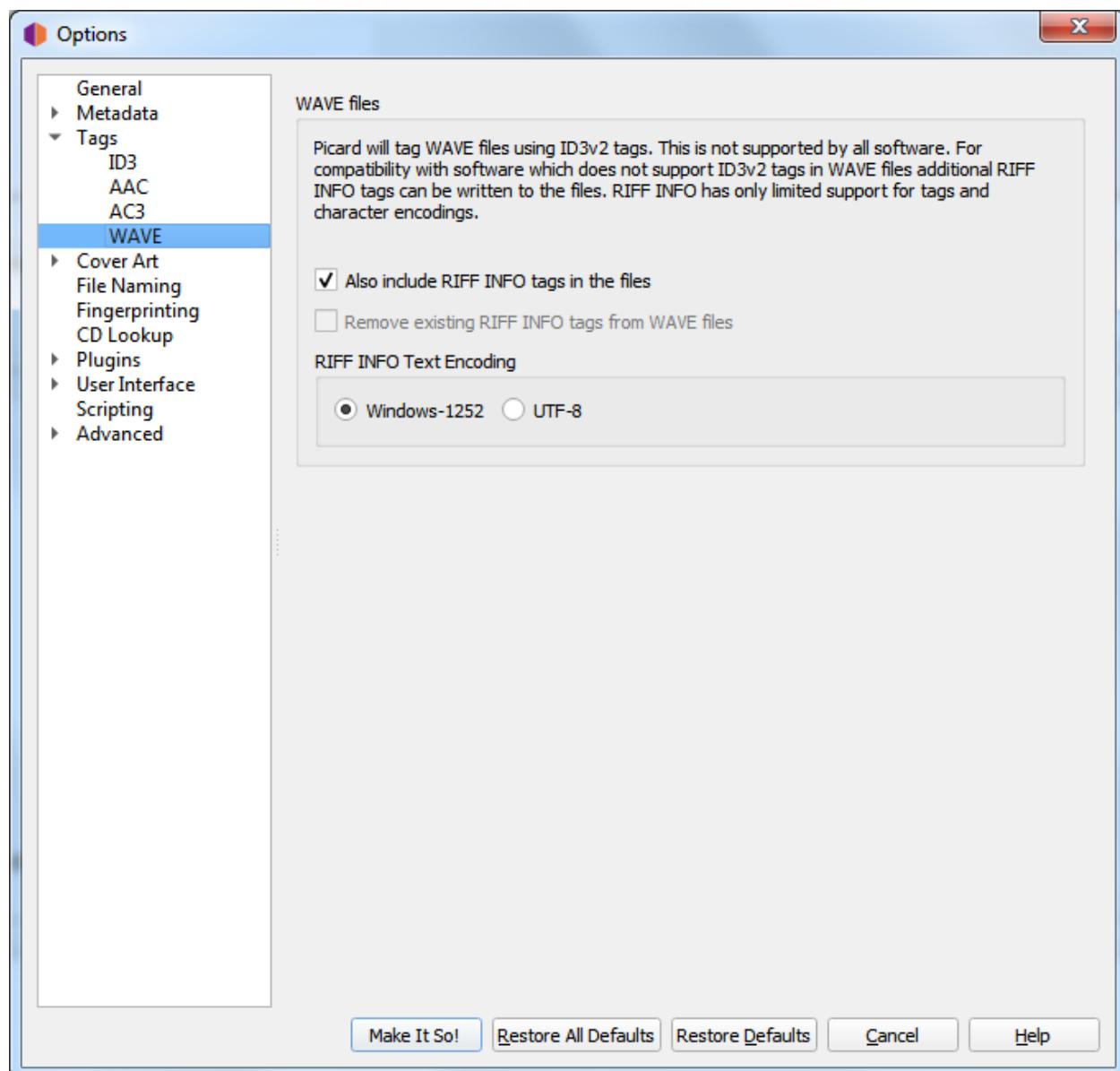
existantes seront conservées dans le fichier.

** Supprimer les balises APEv2 **

Si vous avez activé « Ne pas enregistrer les balises », cocher cette option obligera Picard à supprimer les balises APEv2 existantes du fichier lors de l'enregistrement.

Quelle que soit la façon dont vous avez configuré l'enregistrement des balises, Picard lira toujours les balises APEv2 existantes dans les fichiers AC3.

Fichiers WAVE



WAVE par lui-même en tant que norme ne prend en charge que les balises de bloc INFO, qui sont très limitées. De plus, les blocs INFO n'ont pas de support approprié

pour l'encodage.

Dans tous les cas, Picard écrira des balises ID3 dans les fichiers WAVE. Ceci est soutenu par un certain nombre d'outils; cependant, il n'est pas pris en charge universellement. Les outils ne prenant pas en charge les balises ID3 doivent simplement les ignorer. Si possible, c'est probablement la meilleure option pour avoir des balises appropriées dans les fichiers WAVE. Pour la compatibilité avec les logiciels qui ne prennent pas en charge les balises ID3v2, Picard peut également enregistrer les balises [Resource Interchange File Format](#) (RIFF) INFO tags dans des fichiers WAVE. RIFF INFO n'est lu et écrit qu'en supplément. S'il n'y a pas de balises ID3 existantes, les données de RIFF INFO seront utilisées. Lors de l'enregistrement des fichiers, RIFF INFO sera écrit en plus des balises ID3v2.

L'utilisation par Picard des balises RIFF INFO est déterminée par les paramètres de configuration suivants:

**** Incluez également les balises RIFF INFO dans les fichiers ****

Picard enregistrera les balises RIFF INFO dans les fichiers.

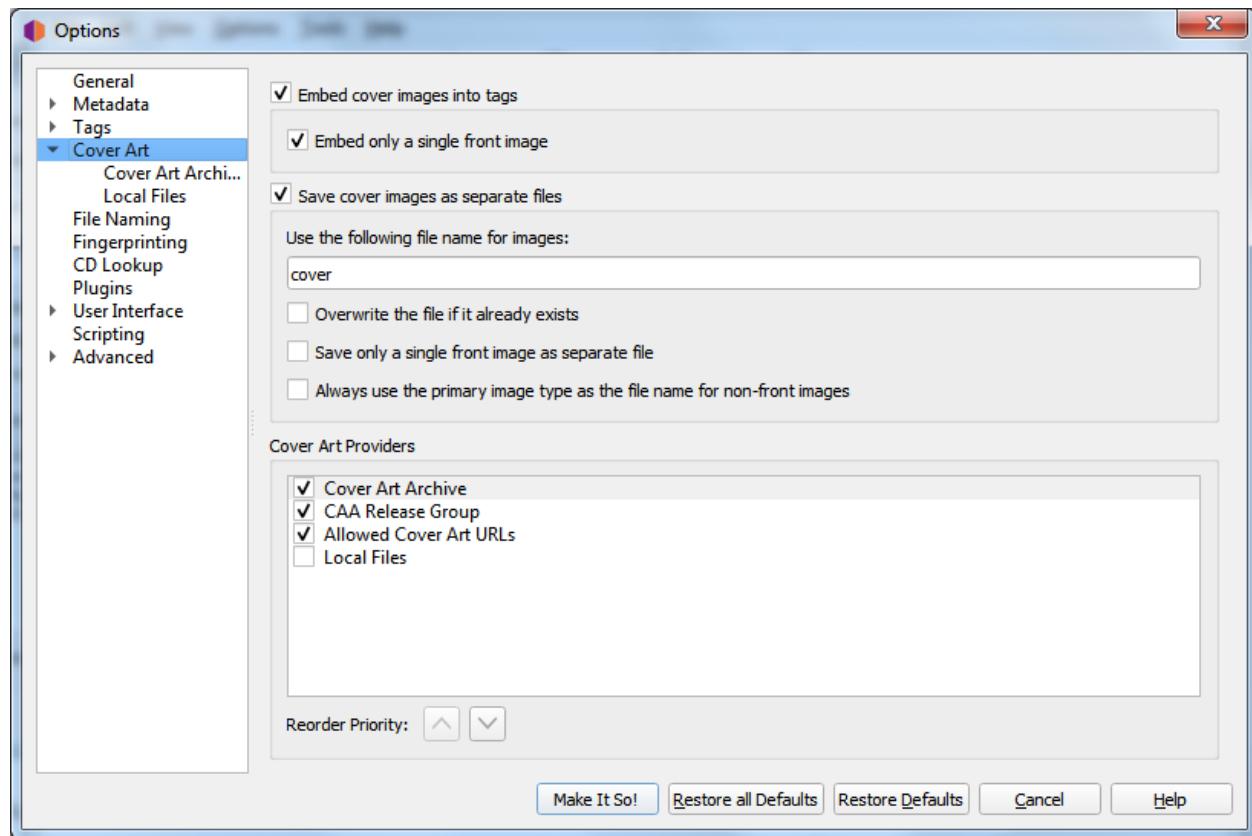
**** Supprimer les balises RIFF INFO existantes des fichiers WAVE ****

Picard supprimera toutes les balises RIFF INFO existantes des fichiers WAVE. Ce paramètre est ignoré si le paramètre précédent est activé pour permettre l'écriture des balises RIFF INFO dans les fichiers.

Encodage de texte RIFF INFO

Ce paramètre vous permet de spécifier l'encodage utilisé pour les balises RIFF INFO écrites. Le paramètre par défaut est le codage Windows-1252. Picard peut également utiliser UTF-8 comme alternative, ce qui permet une prise en charge complète du langage, mais cela dépend du logiciel qui le lit. En règle générale, si le logiciel prend en charge cela, il lira de toute façon les balises ID3, il n'y a donc pas grand chose à gagner.

6.3.4 Options de la pochette



Note: Vous devez activer « Options→Métadonnées→Utiliser les relations de publication » pour que Picard puisse télécharger la pochette à partir des relations de pochette MusicBrainz.

Emplacement

Incorporez des images de couverture dans des tags

Permet aux images d'être intégrées directement dans vos fichiers musicaux. Bien que cela utilisera plus d'espace de stockage que de le stocker en tant que fichier image séparé dans le même répertoire, certains lecteurs de musique afficheront uniquement les images intégrées et ne trouveront pas les fichiers séparés.

Incorporer une seule image de face

N'intègre qu'une seule image avant dans vos fichiers musicaux. Aucune autre image, quel que soit son type, ne sera intégrée. De nombreux lecteurs de musique n'afficheront qu'une seule image intégrée, de sorte que l'incorporation d'images supplémentaires peut ne pas ajouter de fonctionnalité.

Enregistrer les images de couverture dans des fichiers séparés

Dans le masque de nom de fichier, vous pouvez utiliser n'importe quelle variable ou fonction de [Étiquettes Picard](#) et [Fonctions des scripts Picard](#). Le masque ne doit pas contenir d'extension de fichier; celle-ci est ajoutée automatiquement en fonction du type d'image réel. La valeur par défaut est « cover ». Si vous changez cette valeur pour « folder », Windows affichera l'image comme un aperçu du répertoire contenant le fichier.

En plus des variables de script déjà disponibles pour un titre, vous pouvez utiliser les variables suivantes spécifiques au cover art:

- `coverart_maintype`: Le type principal (par exemple: front, back, booklet). Pour les images avant, ce sera toujours « front ».
- `coverart_types`: Liste complète de tous les types affectés à cette image.
- `coverart_comment`: Le commentaire de la pochette.

Par exemple, en spécifiant un masque de dénomination de fichier tel que:

```
%albumartist% - %originalyear% - %album% - %coverart_maintype%
```

précèdera le nom du fichier avec l'artiste de l'album, l'année de sortie de l'original et le titre de l'album.

Vous pouvez également demander à Picard d'enregistrer les images dans un sous-répertoire en l'incluant dans le masque de dénomination des fichiers. Par exemple:

```
Artwork/%albumartist% - %originalyear% - %album% - %coverart_
↪maintype%
```

qui placera les images dans un sous-répertoire appelé « Artwork ».

Écraser le fichier s'il existe déjà

Cochez ceci pour remplacer les fichiers existants. Ceci est particulièrement recommandé si vous essayez d'écrire des aperçus de « folder » pour Windows.

Enregistrer une seule image de face en tant que fichier séparé

Cela indique à Picard de ne sauvegarder que la première image «avant» dans un fichier séparé avec la version. Aucune autre image «avant» ou image d'un autre type ne sera enregistrée. Si cette case n'est pas cochée, toutes les images «avant» seront enregistrées en tant que fichiers séparés, avec tout autre type d'image spécifié à télécharger.

Utilisez toujours le type d'image principal comme nom de fichier pour les images non recto

Ce paramètre modifie la manière dont Picard nomme les fichiers image autres que les images avant.

Lorsqu'elle est cochée, Picard utilisera le type de l'image (par exemple: dos, livret, etc.) comme nom de fichier lors de l'enregistrement, tant que le type n'est pas au recto. Si plusieurs types ont été attribués à l'image, le premier type sera utilisé. Par exemple, si l'image est de types «back» et «raw», alors «back» sera utilisé pour le nom de fichier. Si elle n'est pas cochée ou si l'image est de type «front», Picard utilisera le nom de fichier spécifié dans le paramètre «Utiliser le nom de fichier suivant pour les images».

Fournisseurs d'art de couverture

Picard peut télécharger Cover Art à partir d'un certain nombre de sources, et vous pouvez choisir les sources que vous souhaitez que Picard utilise. Vous pouvez activer plus d'un fournisseur et choisir l'ordre dans lequel les fournisseurs sont interrogés. Picard essaiera les fournisseurs de haut en bas jusqu'à ce qu'une image soit renvoyée.

Archives d'art de couverture

Les Cover Art Archive (CAA) sont les archives MusicBrainz des couvertures en coopération avec [Internet Archive](#). Le Cover Art Archive est la base de données la plus complète des couvertures (par exemple: couvertures avant, arrière, livrets, CD).

Groupe de version CAA

Ce fournisseur utilise l'image de couverture de Cover Art Archive attribuée au groupe de versions. Il s'agit généralement de l'image qui décrit le mieux le groupe de versions dans son ensemble ou l'image avec la meilleure qualité visuelle, mais ce n'est pas nécessairement la couverture exacte de la version que vous marquez. Ce fournisseur est un bon choix si vous vous souciez davantage de la qualité visuelle que d'avoir une représentation exacte de votre version. C'est également une bonne solution de secours pour le fournisseur de Cover Art Archive.

URL de Couverture Autorisées

Cela utilisera des images fournies par des sites tiers approuvés. L'emplacement de l'image est stocké en tant que relation URL pour la version dans la base de données MusicBrainz, et seuls les sites approuvés peuvent être utilisés pour cette relation.

Note: Ce type de relation est désormais obsolète dans MusicBrainz et n'est plus utilisé. Voir [Liste blanche des couvertures](#) dans le Guide de style pour plus d'informations.

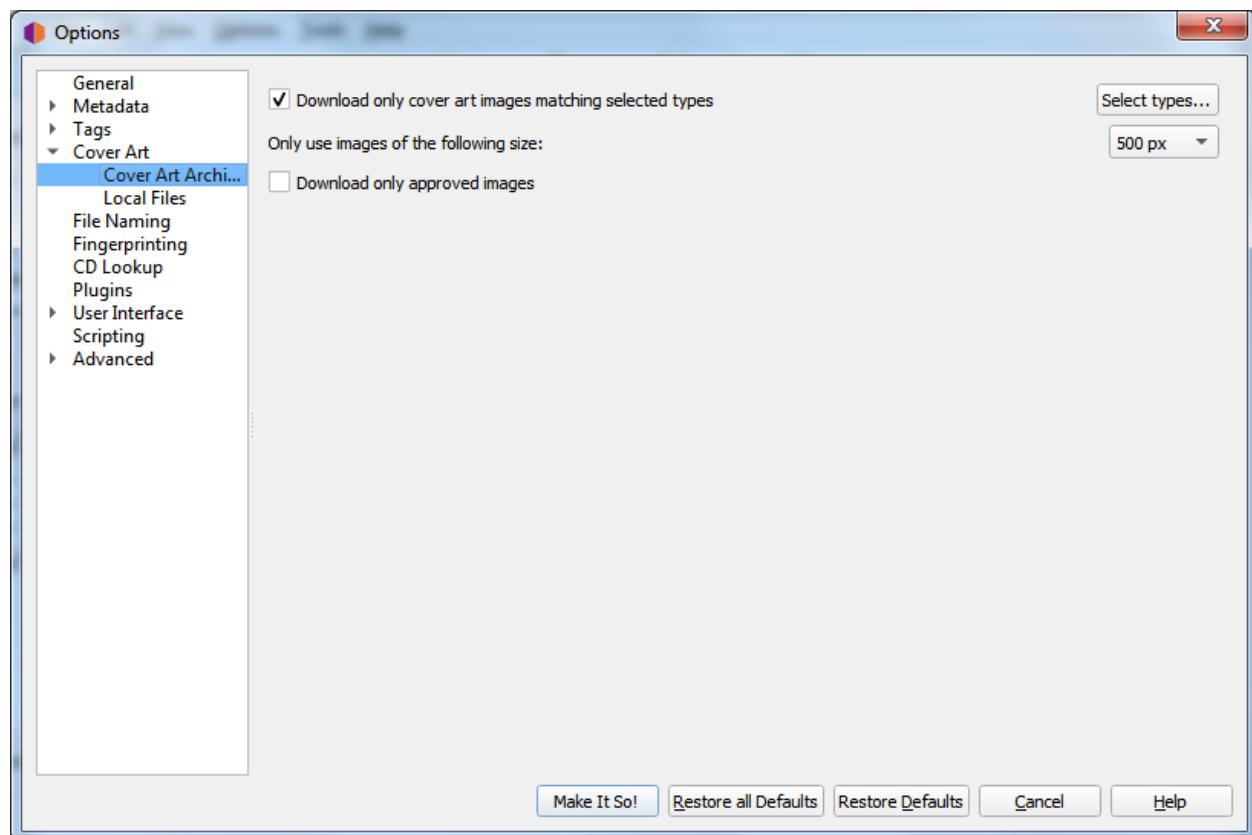
Fichiers locaux

Chargez la pochette à partir de fichiers locaux. Les noms de fichiers à charger peuvent être configurés dans les options du fournisseur [Fichiers locaux](#).

En plus des fournisseurs de couvertures intégrés décrits ci-dessus, des fournisseurs de couvertures supplémentaires peuvent être installés en tant que **plugins**.

- **Amazon**: Amazon a souvent une pochette alors que d'autres sites ne le font pas, cependant, bien que cette illustration soit presque toujours pour le bon artiste et l'album, elle peut ne pas être la pochette absolument correcte pour la version spécifique avec laquelle vous avez tagué ta musique. *Remarque: le fournisseur de couvertures Amazon était intégré à Picard 2.1.3 et aux versions antérieures. Pour les versions ultérieures, il doit être installé en tant que plugin séparé.*
- **fanart.tv**: utilise la pochette de [fanart.tv](#), qui se concentre sur la pochette avec une haute qualité visuelle. Ce fournisseur fournit un représentant de la pochette pour le groupe de versions et non pour la version individuelle.
- **TheAudioDB**: utilise la pochette de [TheAudioDB](#), qui se concentre sur la pochette avec une haute qualité visuelle. Ce fournisseur fournit un représentant de la pochette pour le groupe de versions et non pour la version individuelle.

Cover Art Archive



Dans cette section, vous pouvez décider quels types de pochettes vous souhaitez télécharger à partir des archives de pochettes et quelle qualité (taille) vous voulez télécharger. Évidemment, meilleure est la qualité, plus la taille des fichiers est grande.

Télécharger uniquement les images de couverture correspondant aux types

sélectionnés

Lors de la sélection des types d'images de couverture, vous pouvez sélectionner les types à inclure et à exclure de la liste de téléchargement. Les images CAA avec un type d'image trouvé dans la liste « Inclure » seront téléchargées et utilisées à moins qu'elles n'aient également un type d'image trouvé dans la liste « Exclure ». Les images dont les types se trouvent dans la liste « Exclure » ne seront jamais utilisées. Les types d'images n'apparaissant pas dans les listes « Inclure » ou « Exclure » ne seront pas pris en compte pour déterminer s'il faut ou non télécharger et utiliser une image CAA.

La plupart des lecteurs de musique n'afficheront qu'une seule pièce de couverture pour l'album, et la plupart des gens choisissent Front (couverture) pour cela.

N'utilisez que des images de la taille suivante

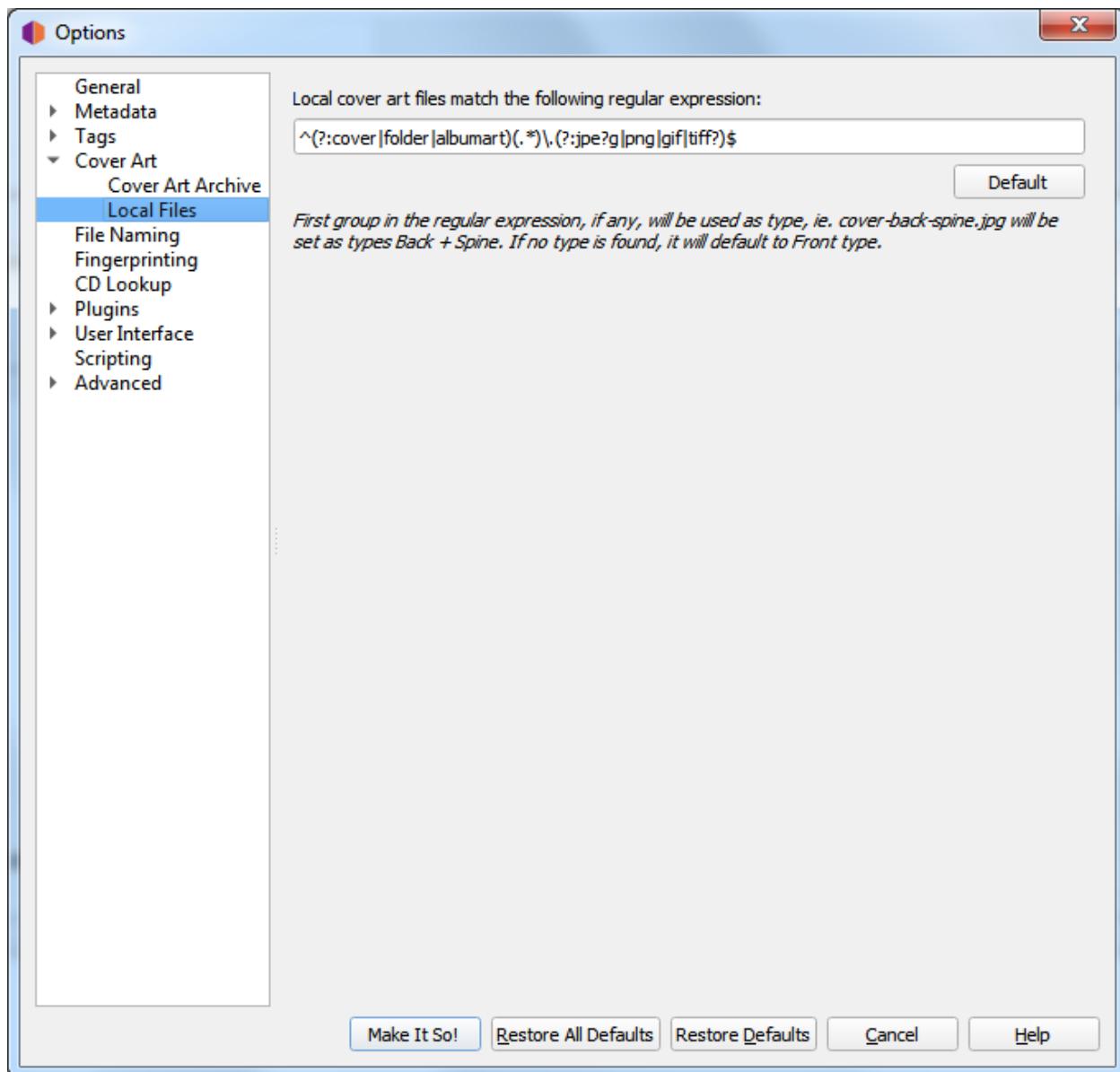
Cela identifie la taille de l'image à télécharger à partir du CAA. Les options sont 250px, 500px, 1200px et pleine taille. Les tailles fixes sont générées automatiquement à partir de l'image pleine taille, à condition qu'elle soit supérieure ou égale à la taille fixe générée. Les images générées sont carrées et remplies selon les besoins si l'image d'origine n'est pas carrée.

Télécharger uniquement les images approuvées

Lorsqu'elle est cochée, Picard ne téléchargera que les images qui ont été approuvées (c'est-à-dire: la modification pour ajouter l'image a été acceptée et appliquée). Pour autoriser l'utilisation d'images issues de modifications en attente, laissez cette option décochée.

Note: Depuis Picard 1.3, vous pouvez également décider d'utiliser ou non l'image du groupe de versions (le cas échéant) si aucune image de face n'est trouvée pour la version. Dans ce cas, la pochette peut ne pas correspondre à la version exacte que vous étiquetez (par exemple: une pochette en vinyle de 1979 peut être utilisée à la place de la réédition du CD Deluxe 2010).

Fichiers locaux



Dans cette section, vous pouvez configurer les noms de fichiers à utiliser par le fournisseur de pochettes de fichiers locaux. Si vous essayez de collecter plus d'une image, la dénomination est importante.

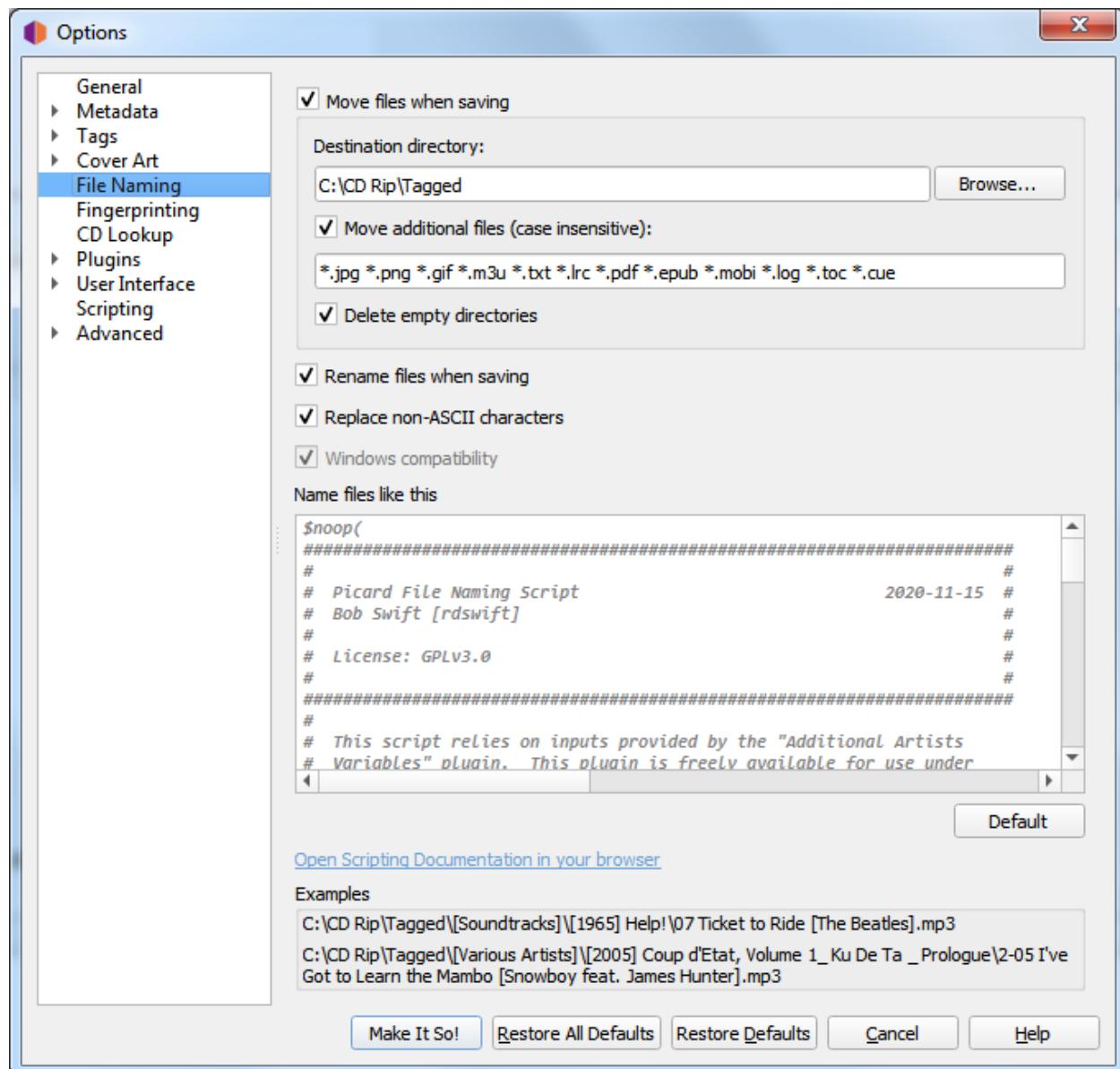
Les noms de fichiers sont définis à l'aide d'une expression régulière. La valeur par défaut est `^(?:cover|folder|albumart)(.*).(?:jpe?g|png|gif|tiff?)$` qui chargera les fichiers avec le nom « cover », « folder » ou « albumart » et l'extension de fichier « jpg », « png », « gif » ou « tiff » (par exemple: « folder.jpg » ou « cover.png »).

La première partie de l'expression régulière est un groupe sans capture: `(?:cover|folder|albumart)`. Les éléments répertoriés dans ce groupe ne seront pas capturés et le type par défaut (Front) s'appliquera.

La deuxième partie de l'expression régulière est un groupe: `(.*)`. C'est la vraie capture, donc si les noms de fichiers correspondent à l'un des types de pochettes, ils seront étiquetés comme tels.

Note: Une erreur courante consiste à ajouter tous les types dans le premier groupe (sans capture). Cela signifie que tous les noms de fichiers normaux seraient jetés dans le type Front et provoqueraient des résultats inattendus.

6.3.5 Options de nom de fichier



Ces options déterminent la manière dont Picard gère les fichiers lorsqu'ils sont enregistrés avec des métadonnées mises à jour.

Déplacer les fichiers lors de l'enregistrement

Si elle est sélectionnée, cette option indique à Picard de déplacer vos fichiers audio vers un nouveau répertoire lorsqu'il les enregistre. Une utilisation pour cela est de garder votre travail organisé: tous les fichiers non étiquetés sont sous « Répertoire A », et lorsque Picard les marque, il les déplace vers « Répertoire B ». Lorsque « Répertoire A » est vide, votre travail de marquage est terminé.

Si cette option n'est pas cochée, Picard laissera les fichiers dans le même répertoire lorsqu'ils seront enregistrés.

Notez que les options « Renommer les fichiers » et « Déplacer les fichiers » sont indépendantes les unes des autres. «Renommer les fichiers» fait référence à Picard changeant les noms de fichiers, généralement en fonction des noms d'artiste et de piste. « Déplacer les fichiers » fait référence au déplacement de fichiers Picard vers de nouveaux répertoires, en fonction d'un répertoire parent et de sous-répertoires spécifiés, généralement basés sur le nom de l'artiste de l'album et le titre de la sortie. Cependant, ils utilisent tous les deux la même «chaîne de dénomination de fichier». « Déplacer les fichiers » utilise la partie vers le haut jusqu'au dernier «/». « Renommer les fichiers » utilise la partie après le dernier «/».

Répertoire de destination

Cela spécifie le répertoire parent de destination vers lequel les fichiers sont déplacés lorsqu'ils sont enregistrés, si l'option « Déplacer les fichiers lors de l'enregistrement » est sélectionnée. Si vous utilisez le répertoire « . » les fichiers seront déplacés par rapport à leur emplacement actuel. S'ils sont déjà dans une sorte de structure de répertoires, cela ne fera probablement pas ce que vous voulez!

Déplacer des fichiers supplémentaires

Entrez des modèles qui correspondent à tous les autres fichiers que Picard doit déplacer lors de l'enregistrement de fichiers musicaux (par exemple: « folder.jpg », « *. png », « *. cue », « *. log »). Les motifs sont séparés par des espaces. Lorsque ces fichiers supplémentaires sont déplacés, ils se retrouvent dans le répertoire de publication avec vos fichiers musicaux. Dans un modèle, le caractère « * » correspond à zéro ou plusieurs caractères. Un autre texte, comme « .jpg », correspond à ces caractères exacts. Ainsi « *.jpg » correspond à « cover.jpg », « liner.jpg », « a.jpg » et « .jpg », mais pas à « nomatch.jpg2 ».

Supprimer les répertoires vides

Lorsqu'il est sélectionné, Picard supprimera les répertoires devenus vides une fois le déplacement terminé. Laissez cette case non cochée si vous voulez que Picard laisse la structure du répertoire source inchangée. Cochez cette case peut être pratique si vous utilisez l'option « déplacer les fichiers » pour organiser votre travail. Un répertoire vide n'a plus de travail à faire, et la suppression du répertoire le clarifie.

Renommer les fichiers lors de l'enregistrement

Sélectionnez cette option pour permettre à Picard de modifier les noms de fichier et de répertoire de vos fichiers lorsqu'il les enregistre, afin de rendre les noms de fichier et de répertoire cohérents avec les nouvelles métadonnées.

Remplacer les caractères non ASCII

Sélectionnez cette option pour remplacer les caractères non ASCII par leur équivalent ASCII (par exemple: "á", "ä" et "ă" par "a"; "é", "ë" et "ë" par "e"; "æ" avec « ae »). Plus d'informations sur les caractères ASCII peuvent être trouvées sur [Wikipedia](#).

Compatibilité Windows

Cette option indique à Picard de remplacer tous les caractères incompatibles avec Windows par un trait de soulignement. Ceci est activé par défaut sur les systèmes Windows, sans option de désactivation.

Nommez des fichiers comme celui-ci

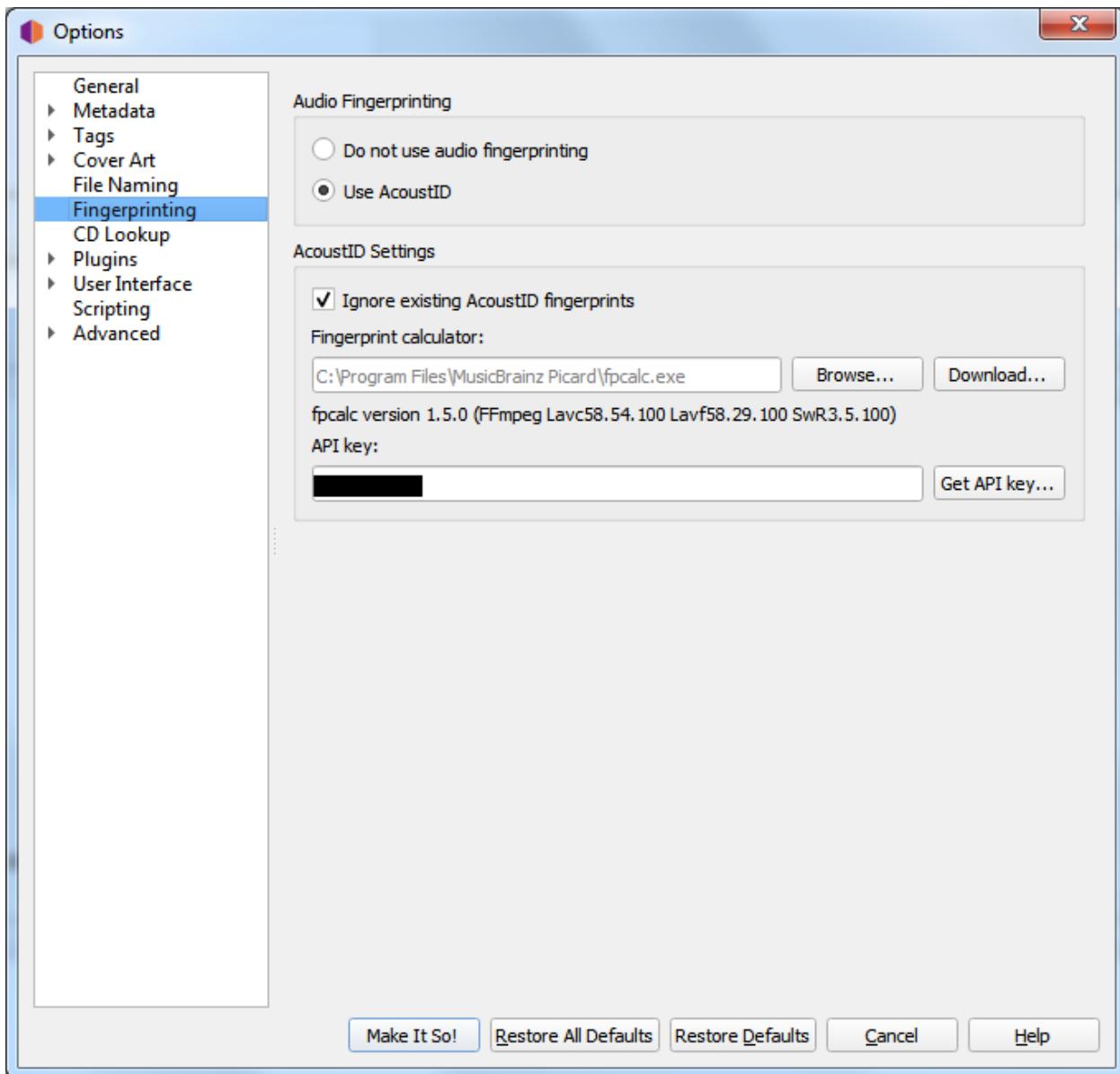
Une zone d'édition qui contient une chaîne de mise en forme qui indique à Picard quel doit être le nouveau nom du fichier et ses répertoires contenant en termes de valeurs de métadonnées diverses. La chaîne de formatage est dans [langage de script de Picard](#) où le texte bleu foncé commençant par un "\$" est un [nom de la fonction](#) et noms en bleu clair entre les signes "%" sont ceux de Picard [tag et noms de variable](#), et est généralement appelé « script de nommage de fichier ». Notez que l'utilisation d'un "/" dans la chaîne de formatage sépare le répertoire de sortie du nom de fichier. La chaîne de mise en forme peut contenir n'importe quel nombre de caractères "/". Tout ce qui précède le dernier "/" est l'emplacement du répertoire, et tout ce qui se trouve après le dernier "/" devient le nom du fichier.

Il n'y a qu'un seul script de dénomination de fichier défini dans les paramètres d'un utilisateur, bien qu'il puisse varier d'un simple script sur une ligne tel que `%album%/%title%` à un script très complexe utilisant différents formats de dénomination de fichier basés sur différents Critères. Dans tous les cas, les fichiers seront enregistrés en utilisant le texte sorti par le script.

Les scripts sont souvent abordés dans le [MetaBrainz Community Forum](#), et il existe un fil de discussion spécifique à [la dénomination de fichier et les extraits de script](#).

Note: Les nouveaux ensembles de balises ou les balises modifiées par le script de dénomination de fichier ne seront pas écrites dans les métadonnées des fichiers de sortie.

6.3.6 Options d'empreintes digitales



Si vous sélectionnez un fichier ou un cluster dans la partie gauche de l'écran de Picard et sélectionnez « *Outils → Scan* », Picard invoquera un programme pour scanner les fichiers et produire une empreinte pour chacun qui pourra alors être utilisé pour rechercher le fichier sur MusicBrainz.

MusicBrainz ne prend actuellement en charge que [AcoustID](#) (un système Open Source [acoustic fingerprinting](#) créé par Lukáš Lalinský) mais a précédemment pris en charge TRM et MusicID PUID.

Empreinte audio

Cela vous permet de choisir d'activer ou non la prise d'empreintes acoustiques dans Picard. Si l'empreinte digitale acoustique est désactivée,

toutes les options restantes de cet onglet seront verrouillées et ignorées.

Ignorer les empreintes digitales AcoustID existantes

Lorsqu'elle est cochée, les informations d'empreinte digitale AcoustID existantes ne seront pas utilisées et les fichiers seront à nouveau analysés.

Calculateur d'empreintes digitales

Cela identifie le programme externe de votre système qui sera utilisé pour calculer les empreintes digitales AcoustID. Par défaut, Picard utilise l'utilitaire [Chromaprint \(fpcalc\)](#) qui est inclus avec l'installation de Picard.

Picard détectera automatiquement le chemin à moins que vous ne l'ayez spécifiquement écrasé avec quelque chose de différent du chemin détecté. « *Options → Empreinte digitale* » affichera le chemin détecté automatiquement comme un indicateur d'espace réservé dans l'entrée de texte, et l'utilisera également pour valider l'exécutable. L'utilisateur n'a besoin de sélectionner réellement quelque chose que si la détection automatique ne fonctionne pas. Si l'utilisateur a sélectionné un chemin spécifique, celui-ci sera utilisé.

Clé API

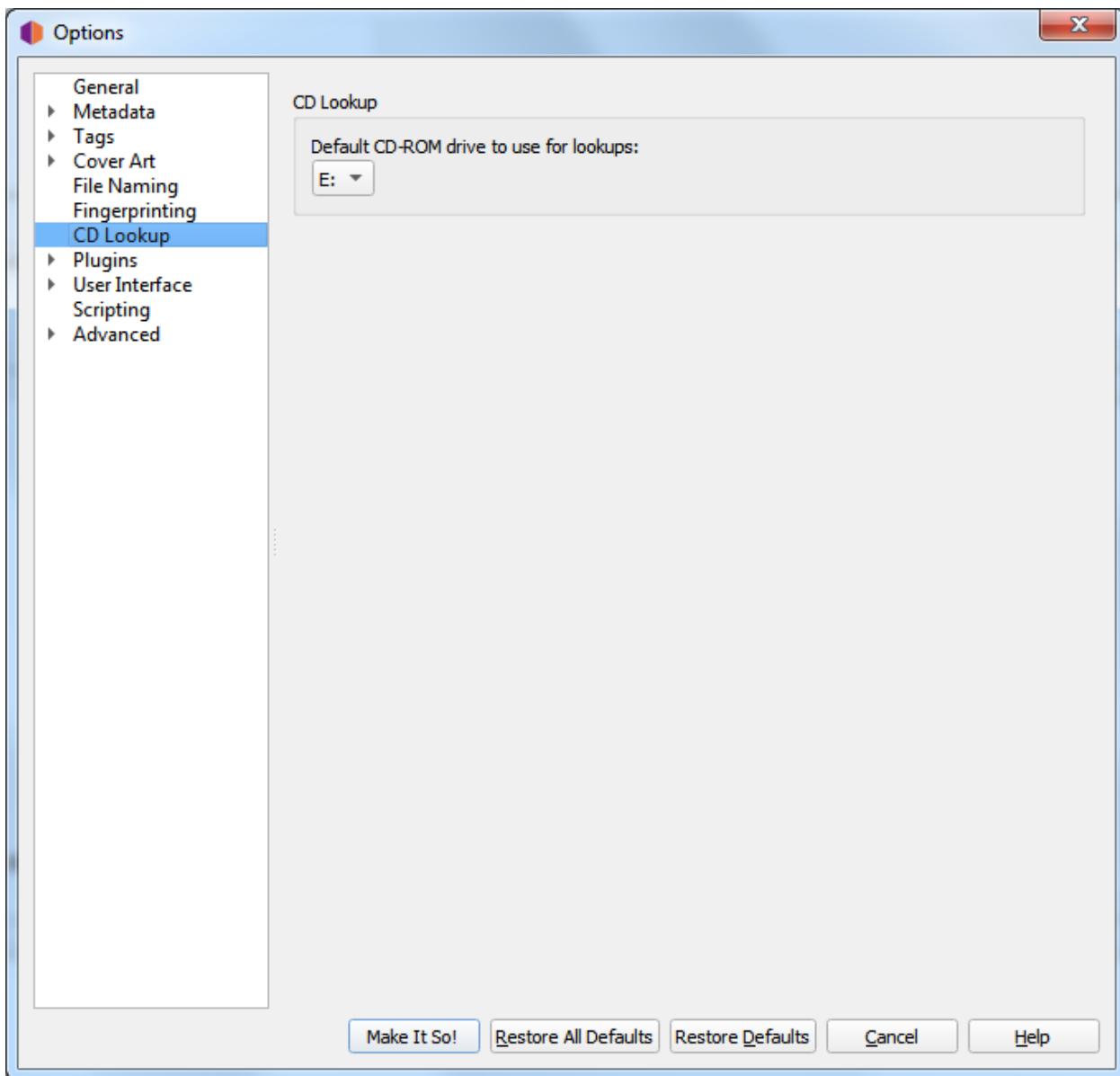
Clé utilisée pour accéder à l'API AcoustID pour rechercher et soumettre des empreintes digitales AcoustID. Il n'y a aucun coût pour obtenir une clé API.

6.3.7 Options de recherche de CD

Cette section vous permet de sélectionner le périphérique de CD-ROM à utiliser par défaut pour rechercher un CD.

Sur les systèmes Windows et Linux, vous pouvez annuler ce paramètre en cliquant sur « *Outils → Rechercher un CD...* » et en sélectionnant le périphérique souhaité dans la liste des périphériques disponibles.

Windows



Sous Windows, Picard a un menu déroulant répertoriant les différents lecteurs de CD qu'il a trouvés. Déroulez le menu et sélectionnez le lecteur que vous souhaitez utiliser par défaut.

Vous pouvez annuler ce paramètre en cliquant sur :menuselection:"Outils --> Rechercher un CD..." et en sélectionnant le périphérique souhaité dans la liste des périphériques disponibles.

macOS

Dans macOS, l'option de recherche de CD est actuellement un champ de texte. Le périphérique est généralement /dev/rdisk1.

Si cela ne fonctionne pas, une façon est de simplement continuer à augmenter le nombre (par exemple /dev/rdisk2) jusqu'à ce que cela fonctionne. Une méthode moins d'essai et d'erreur consiste à ouvrir « Terminal » et à taper « mount ». La sortie doit inclure une ligne telle que:

```
/dev/disk2 on /Volumes/Audio CD (local, nodev, nosuid, read-only)
```

Vous devez remplacer /dev/disk par /dev/rdisk, donc si, par exemple, il dit /dev/disk2, vous devez entrer /dev/rdisk2 dans les préférences de Picard.

Linux

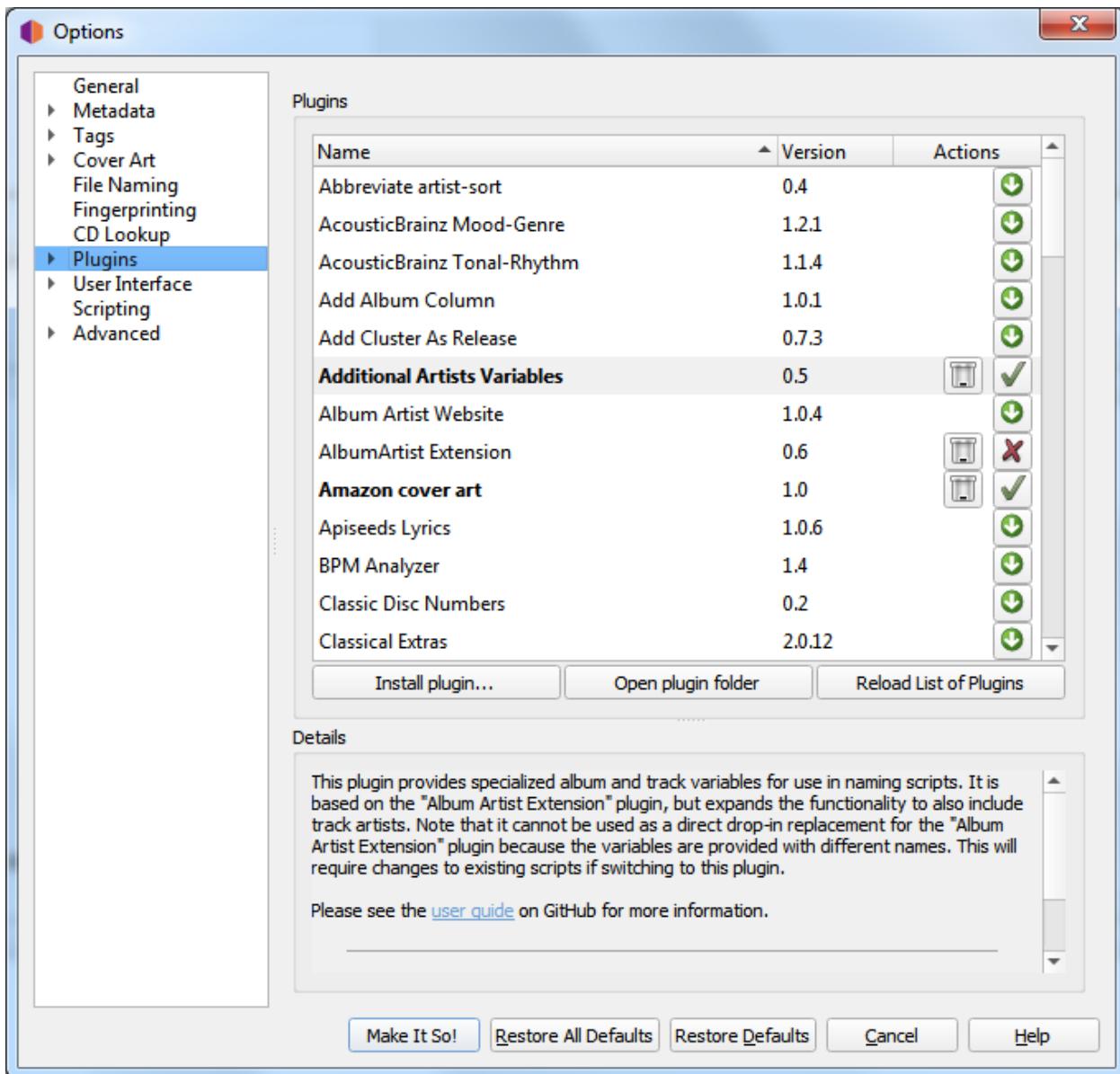
Sous Linux, Picard dispose d'un menu déroulant comme sous Windows pour l'option Recherche de CD. Si vous utilisez une ancienne version de Picard avec un champ texte, vous devez entrer le nom du périphérique (typiquement /dev/cdrom).

Vous pouvez annuler ce paramètre en cliquant sur « *Outils → Rechercher un CD...* » et en sélectionnant le périphérique souhaité dans la liste des périphériques disponibles.

Autres plateformes

Sur d'autres plateformes, l'option de recherche de CD est un champ de texte et vous devez entrer le chemin d'accès au lecteur de CD ici.

6.3.8 Plugins Options



Cette section vous permet de gérer les plugins utilisés par Picard. Vous pouvez installer de nouveaux plugins ou activer, désactiver ou désinstaller des plugins actuellement installés. Picard fournit une liste des plugins qui ont été soumis au projet. Une liste des plugins standards est disponible sur la [page des plugins](#) sur le site Picard.

Il existe également un certain nombre de plugins disponibles par des développeurs tiers. Souvent, ceux-ci sont discutés sur le [Forum de discussion communautaire](#) donc si vous recherchez une amélioration ou une fonctionnalité particulière, une recherche peut être utile. De plus, l'un des utilisateurs de MusicBrainz, [Colby Ray](#) maintient une liste non officielle des plugins disponibles sur une [page wiki](#). Les instructions concernant l'installation de plugins tiers sont incluses dans la section « [Installation de plugins tiers](#) » ci-dessous.

Liste des plugins

L'écran affiche une liste des plugins standard et de tous les autres qui ont été installés. Chaque plugin est affiché sur une ligne distincte indiquant le numéro de version et une ou plusieurs icônes d'état / d'action. Les icônes sont:



Cette icône indique que le plugin n'est pas installé. Cliquez sur l'icône pour télécharger et installer le plugin.



Cette icône indique qu'une version plus récente du plugin est disponible. Cliquez sur l'icône pour télécharger et installer la version mise à jour.



Cette icône indique que le plugin est installé et actuellement activé. Cliquer sur l'icône désactivera le plugin, mais il sera toujours installé.

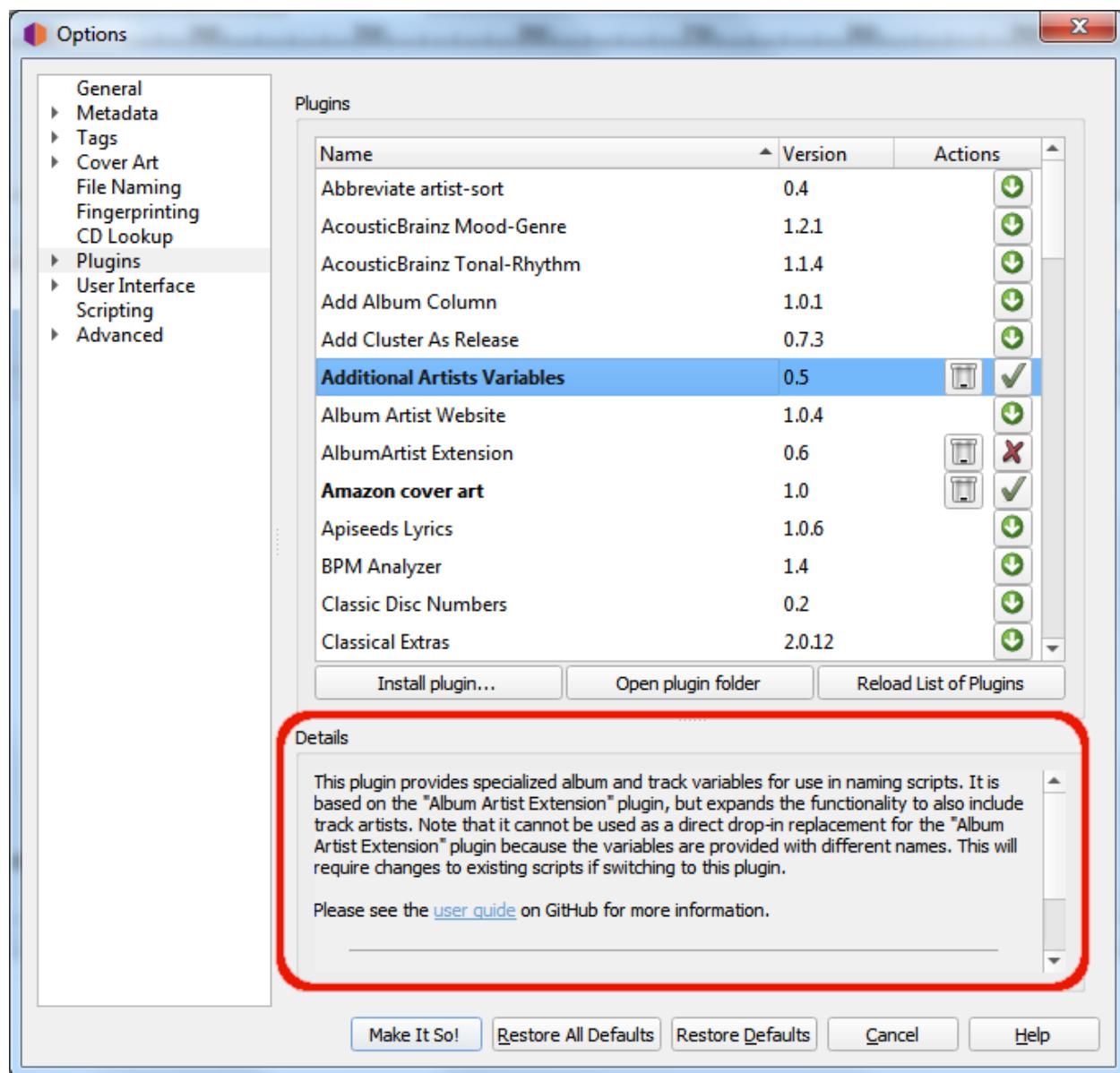


Cette icône indique que le plugin est installé mais actuellement désactivé. Cliquez sur l'icône pour activer le plugin.



Cette icône indique que le plugin est actuellement installé. Cliquez sur l'icône pour désinstaller le plugin.

Lorsqu'un plugin dans la liste est sélectionné (c'est-à-dire: mis en évidence), une brève description du plugin sera affichée dans la section « Détails » sous la liste.

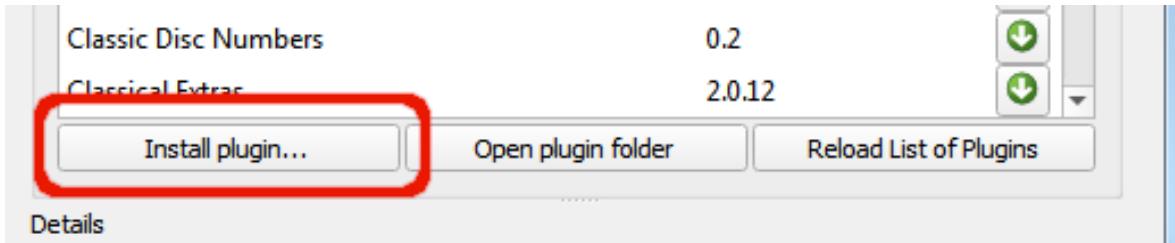


Note: Certains plugins ont leur propre page d'options qui apparaîtra généralement sous la section « Plugins » des options.

Installation de plugins tiers

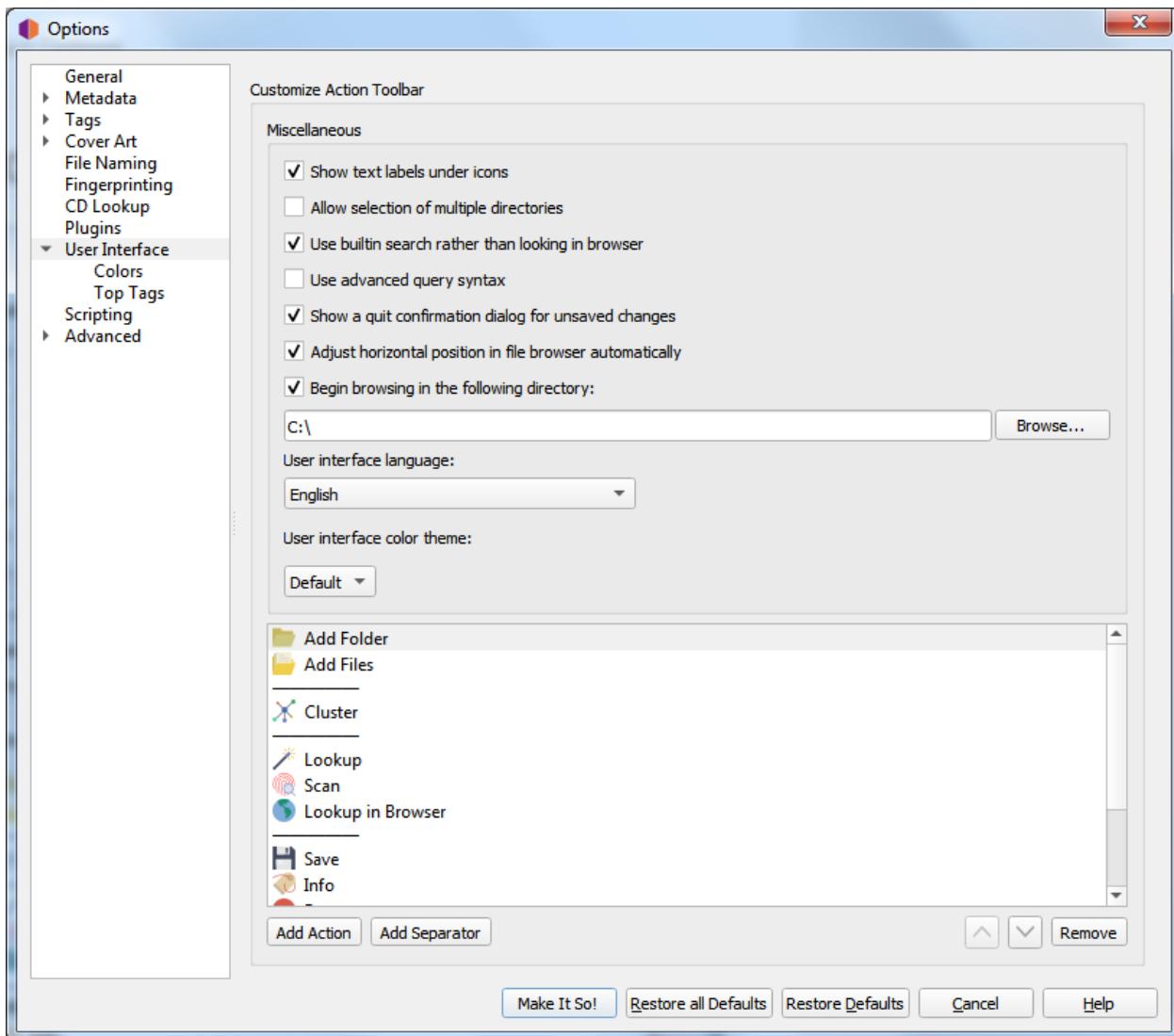
Vous pouvez installer un plugin tiers qui n'apparaît pas dans la liste des plugins, en suivant les étapes suivantes:

1. Téléchargez le plugin et enregistrez-le sur un lecteur local.
2. Sélectionnez l'action « Installer le plugin... », située juste en dessous de la liste des plugins.



3. Accédez au fichier que vous avez téléchargé à l'étape 1 et sélectionnez-le. Le fichier sera copié dans le dossier du plugin et apparaîtra dans la liste des plugins.
4. Activez le plugin si vous le souhaitez et sélectionnez « Make It So! » bouton d'action en bas de la fenêtre.

6.3.9 Options de l'interface utilisateur



Afficher les étiquettes de texte sous l'icône

Si cette option est désactivée, les étiquettes de texte sous les icônes de la barre d'outils ne seront pas affichées, ce qui fait que la barre d'outils apparaîtra un peu plus petite.

Autoriser la sélection de plusieurs répertoires

L'activation de cette option contournera le sélecteur de répertoire natif et utilisera la boîte de dialogue de fichier de Qt. Cela peut être souhaitable car le sélecteur de répertoire natif ne vous permet généralement pas de sélectionner plus d'un répertoire. Ceci s'applique à la boîte de dialogue « *Fichier → Ajouter un dossier* ». Le navigateur de fichiers permet toujours la sélection de plusieurs répertoires.

Utilisez la recherche intégrée plutôt que de rechercher dans le navigateur

Lorsque cette option est activée, la recherche d'albums, d'artistes ou de pistes affichera les résultats dans une boîte de dialogue. Par défaut, cette option est activée. Si cette option est désactivée, Picard ouvrira une recherche sur MusicBrainz.org dans votre navigateur Web par défaut.

Utilisez la syntaxe de requête avancée

Cela activera l'analyse de la syntaxe de requête avancée sur vos recherches. Cela s'applique uniquement à la zone de recherche en haut à droite de Picard, pas aux boutons de recherche.

Afficher une boîte de dialogue de confirmation de fermeture pour les modifications non enregistrées

Lorsque cela est activé, Picard affichera une boîte de dialogue lorsque vous essayez de quitter le programme avec des fichiers non enregistrés chargés. Cela peut aider à éviter de perdre accidentellement les modifications de balises que vous avez apportées, mais pas encore enregistrées.

Commencez à naviguer dans le répertoire suivant

Par défaut, Picard se souvient du dernier répertoire utilisé pour charger les fichiers. Si vous activez cette option et fournissez un répertoire, Picard démarrera toujours dans le répertoire fourni.

Langue de l'interface utilisateur

Par défaut, Picard s'affichera dans la langue affichée par votre système d'exploitation, mais vous pouvez le remplacer et sélectionner une autre langue si nécessaire.

Thème de couleurs de l'interface utilisateur

Cette option permet à l'utilisateur de sélectionner le thème de couleurs utilisé par Picard. Sur les systèmes macOS et Windows, les choix disponibles sont les suivants:

- Défaut - Le schéma de couleurs par défaut basé sur les paramètres d'affichage du système d'exploitation.
- Light - Un thème d'affichage lumineux.
- Dark - Un thème d'affichage sombre.

Sous Linux et les systèmes d'exploitation similaires, les choix disponibles sont les suivants:

- Défaut - Le schéma de couleurs par défaut basé sur les paramètres d'affichage du système d'exploitation.
- System - Le thème Qt5 configuré dans l'environnement de bureau.

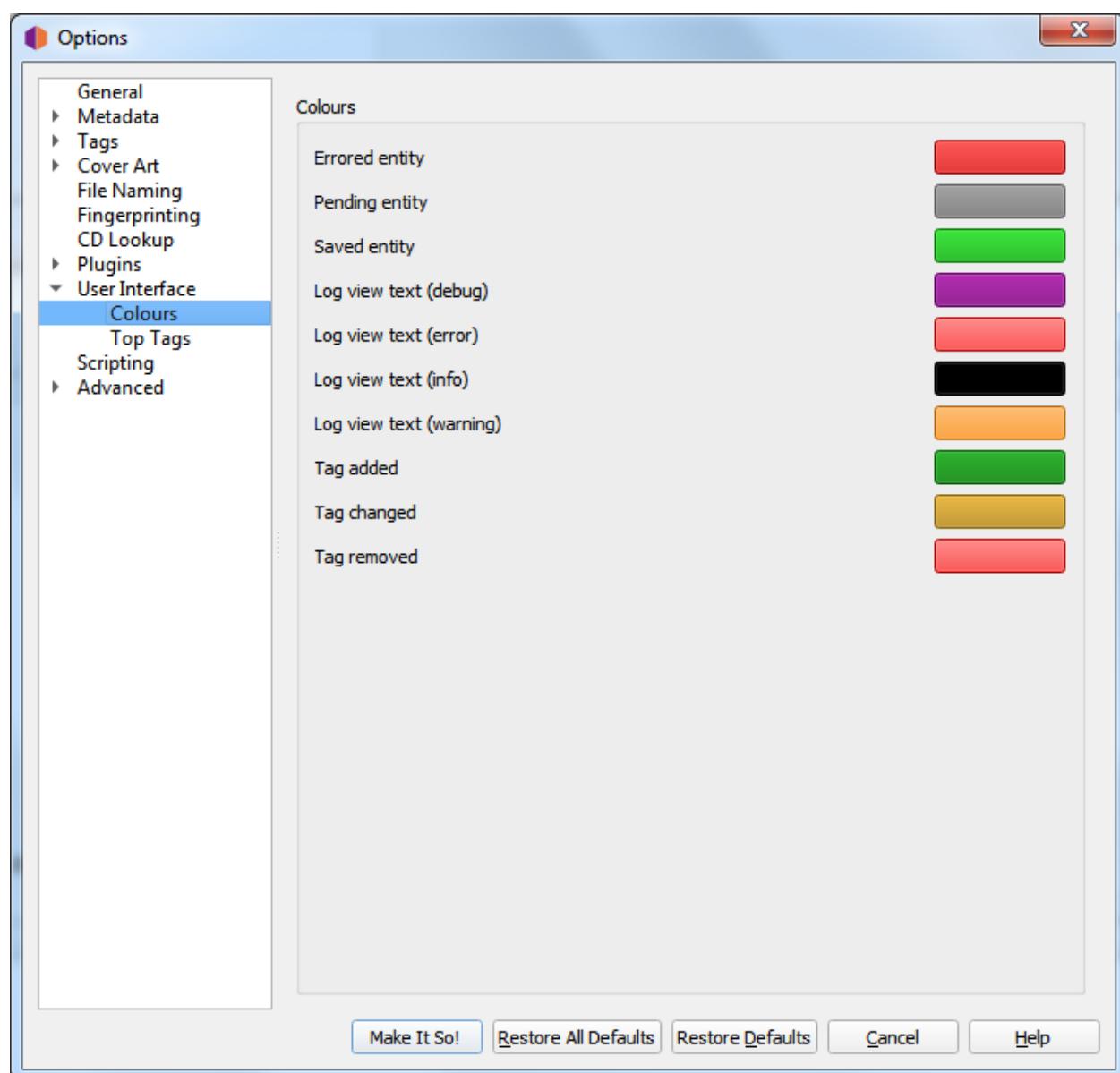
Note: Les couleurs des thèmes clair et foncé peuvent être personnalisées dans la section [Couleurs](#). Des ensembles distincts de sélections de couleurs

sont gérés pour les thèmes clairs et foncés. Les couleurs du thème actuellement affiché sont celles qui sont affichées pour l'édition.

Personnaliser la barre d'outils d'action

Cela vous permet d'ajouter, de supprimer ou de réorganiser les éléments affichés dans la barre d'outils Action.

Couleurs



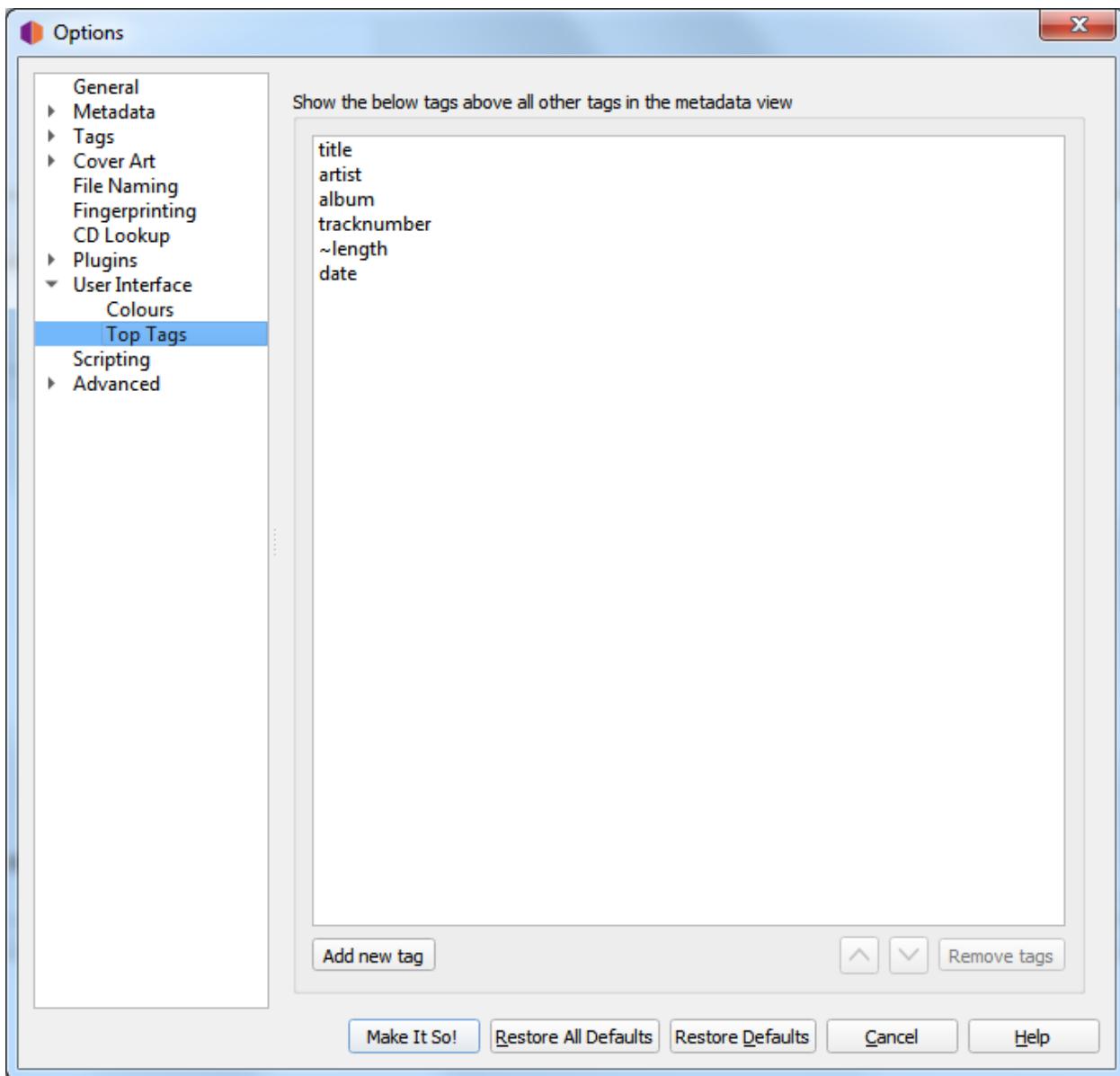
Cette section vous permet de personnaliser les différentes couleurs utilisées dans l'interface utilisateur de Picard. Pour changer une couleur, cliquez simplement sur le bloc de couleur actuellement affiché pour la condition de texte souhaitée pour faire

apparaître une boîte de dialogue de sélection, puis choisissez la couleur souhaitée. Les couleurs peuvent être modifiées pour les conditions de texte suivantes:

- **Entité erronée:** fichiers et autres éléments avec des erreurs de chargement ou d'enregistrement
- **Entité en attente:** fichiers et autres éléments mis en file d'attente pour traitement
- **Entité enregistrée:** fichiers enregistrés avec succès
- **Texte de la vue du journal (débogage):** messages de débogage dans le journal des erreurs / débogage
- **Texte de la vue du journal (erreur):** messages d'erreur dans le journal des erreurs / débogage
- **Texte de la vue du journal (info):** messages d'information dans le journal des erreurs / débogage
- **Texte de la vue du journal (avertissement):** messages d'avertissement dans le journal des erreurs / débogage
- **Balise ajoutée:** nouvelles balises ajoutées dans le volet des métadonnées
- **Balise modifiée:** balises modifiées dans le volet des métadonnées
- **Balise supprimée:** balises supprimées dans le volet des métadonnées

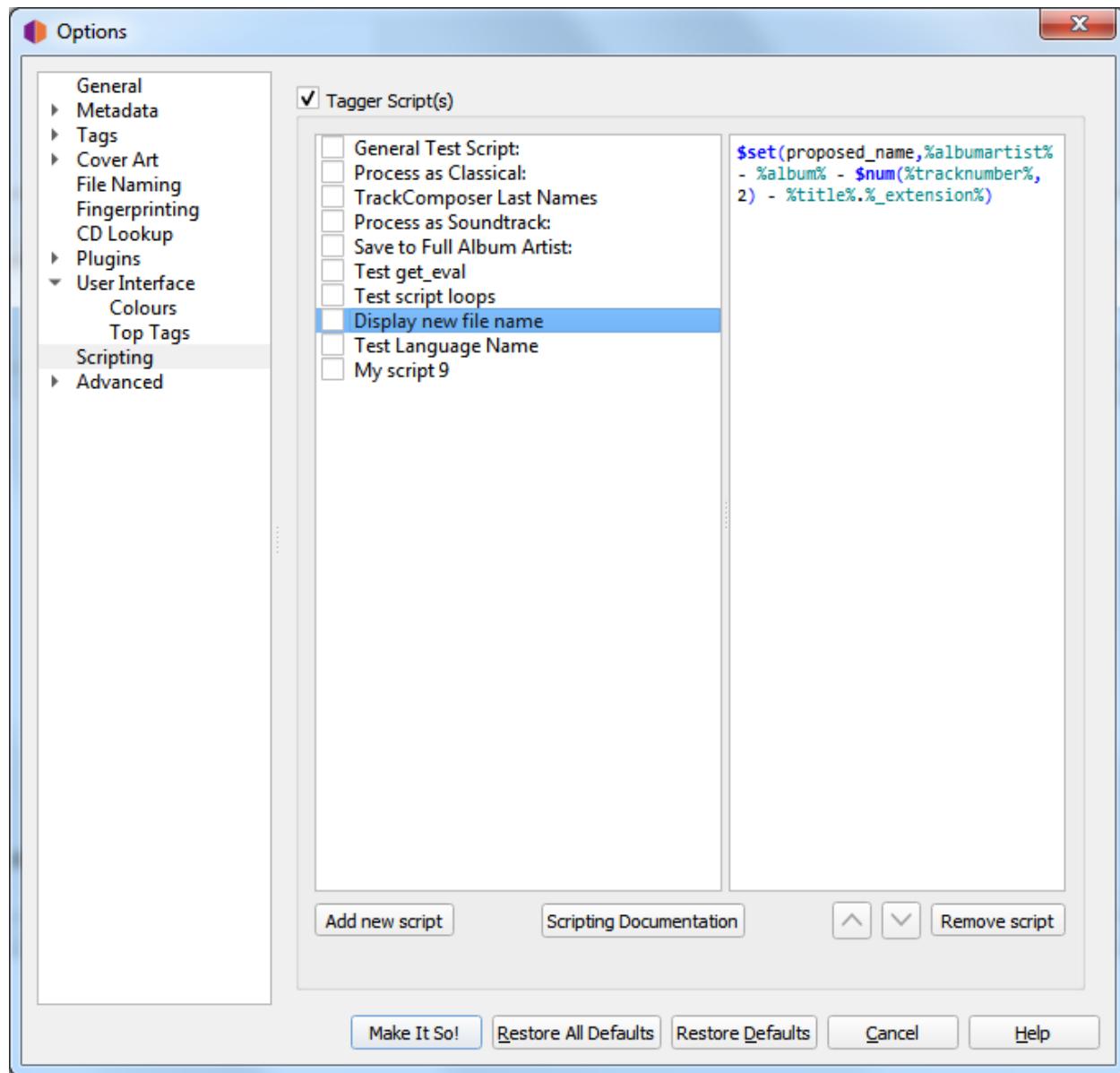
Note: Des ensembles distincts de sélections de couleurs sont gérés pour les thèmes clairs et foncés. Les couleurs du thème actuellement affiché sont celles qui sont affichées pour l'édition.

Top Tags



Les balises spécifiées dans ce paramètre d'option seront toujours affichées dans l'ordre spécifié en haut du volet des métadonnées (qui affiche les métadonnées des fichiers ou pistes sélectionnés). Cela vous permet d'avoir les balises les plus importantes toujours en haut de la liste. Les balises non répertoriées ici seront affichées par ordre alphabétique sous les balises supérieures.

6.3.10 Options de script



Cette section permet la gestion des scripts de balisage définis par l'utilisateur.

La case à cocher « Tagger Script(s) » en haut de la page vous permet de désactiver complètement tous les scripts de balisage. Cela peut être utile lors de la recherche d'un problème avec la configuration de Picard.

Sous la case à cocher se trouvent deux colonnes affichant la liste des scripts dans la colonne de gauche, avec le contenu du script sélectionné affiché dans la colonne de droite. Cette section vous permet d'ajouter, de supprimer et de réorganiser les scripts, d'activer ou de désactiver des scripts individuels, ainsi que de modifier le script actuellement sélectionné.

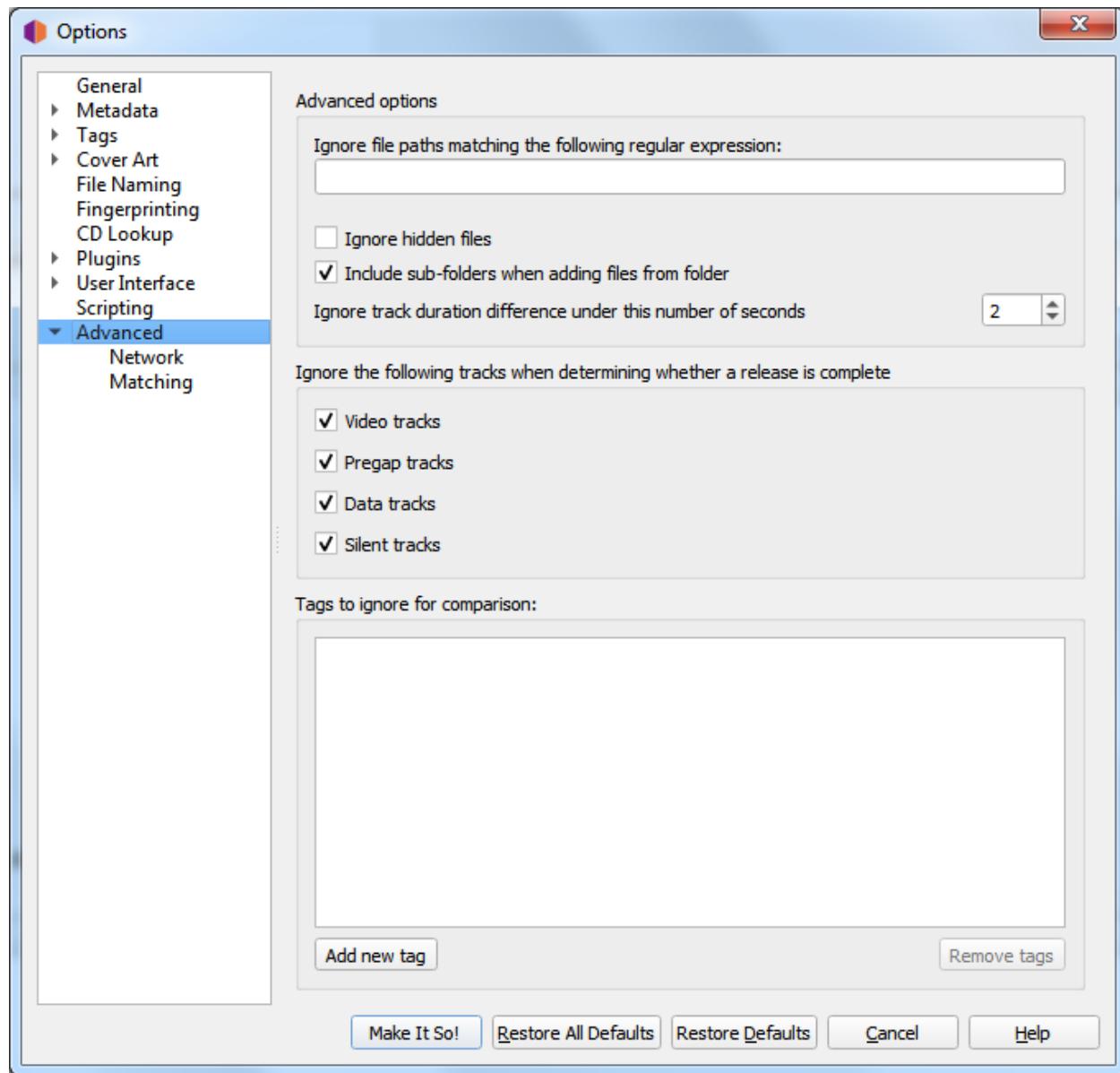
Lorsque la case à cocher à côté du script est cochée, ce script sera exécuté

automatiquement, une fois pour chaque piste de la version, lorsque Picard récupérera des informations pour une version sur le site Web de MusicBrainz. Si la case n'est pas cochée, le script ne sera pas exécuté automatiquement.

Que le script soit exécuté automatiquement ou non, il peut également être exécuté manuellement en cliquant avec le bouton droit de la souris sur un élément dans le volet de clustering (volet central) ou dans le volet de balisage (volet de droite) et en le sélectionnant dans la liste affichée lorsque « *Exécuter les scripts* » est sélectionné. Si un cluster est sélectionné dans le volet central ou si une version est sélectionnée dans le volet droit, le script sera exécuté pour chaque piste du cluster ou de la version sélectionnée. Si une seule piste ou un seul fichier est sélectionné, le script ne sera exécuté que pour cette piste ou ce fichier.

Pour plus d'informations sur les scripts, veuillez consulter les sections « *Scripts* » et « *Scripting* », ainsi que « *Balises & Variables* ».

6.3.11 Options avancées



Ignorer les chemins de fichiers correspondant à l'expression régulière suivante

Vous pouvez spécifier des modèles pour les fichiers et répertoires que Picard ne doit jamais charger. Par exemple, si vous définissez ceci sur l'expression régulière `\.bak$` tout fichier se terminant par « `.bak` » sera ignoré lors du chargement des fichiers.

Ignorer les fichiers cachés

Si cette option est activée, les fichiers et répertoires cachés ne seront pas chargés. Cela inclut également tout fichier ou sous-répertoire dans un répertoire caché.

Inclure les sous-dossiers lors de l'ajout de fichiers à partir de dossiers

Si cette option est activée, Picard chargera tous les fichiers audio dans le répertoire sélectionné et tous ses sous-répertoires. Si elle est désactivée, seuls les fichiers audio du répertoire sélectionné seront chargés.

Ignorer la différence de durée de la piste sous ce nombre de secondes

Ceci spécifie le nombre de secondes pendant lesquelles un fichier peut différer en longueur de la longueur dans la base de données MusicBrainz et toujours être considéré comme identique. La valeur par défaut est de 2 secondes.

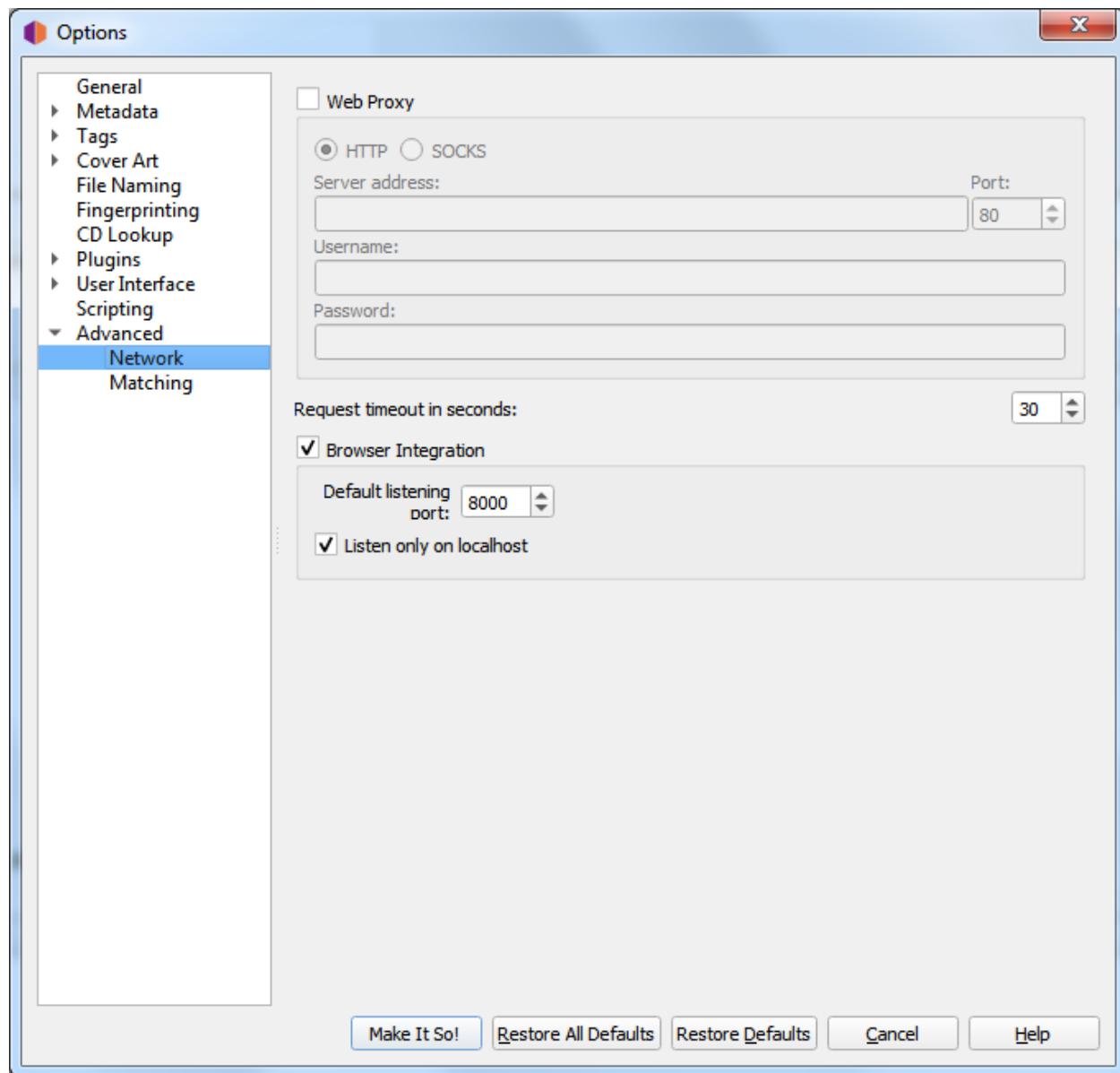
Ignorez les pistes suivantes pour déterminer si une version est terminée

Les pistes manquantes du type sélectionné (c'est-à-dire: vidéo, pré-séquence, données ou silence) seront ignorées lors de la détermination si une sortie est considérée comme terminée. Par exemple, si « vidéo » est sélectionné, une version avec une vidéo bonus sera marquée comme terminée si toutes les pistes audio correspondent à un fichier même si le fichier vidéo est manquant.

Balises à ignorer pour comparaison

Les balises de cette liste ne seront pas prises en compte lors de la comparaison des métadonnées de fichier existantes avec les données extraites de MusicBrainz. Si la seule différence entre les métadonnées du fichier et les métadonnées extraites de MusicBrainz est une balise répertoriée dans cette liste ignorée, le fichier sera considéré comme non modifié.

Réseau



Proxy Web

Si vous avez besoin d'un proxy pour établir une connexion réseau externe, vous pouvez en spécifier un ici. Vous pouvez choisir entre le proxy HTTP et SOCKS. Les paramètres requis sont **Adresse du serveur** et **Port**. Si le proxy requiert une authentification, saisissez également **Nom d'utilisateur** et **Mot de passe**.

Demander un délai d'expiration en secondes

Par défaut, Picard abandonnera les requêtes réseau en cours après 30 secondes d'inactivité. Si nécessaire, vous pouvez modifier le délai d'expiration ici.

Intégration du navigateur

L'intégration du navigateur vous permet de charger des versions et des enregistrements dans Picard directement à partir du site Web MusicBrainz. Une fois que vous avez ouvert musicbrainz.org dans votre navigateur depuis Picard, le site Web affichera le bouton tagger vert  à côté des sorties et des enregistrements. Cliquer sur ce bouton chargera la version ou l'enregistrement correspondant dans Picard.

Port d'écoute par défaut

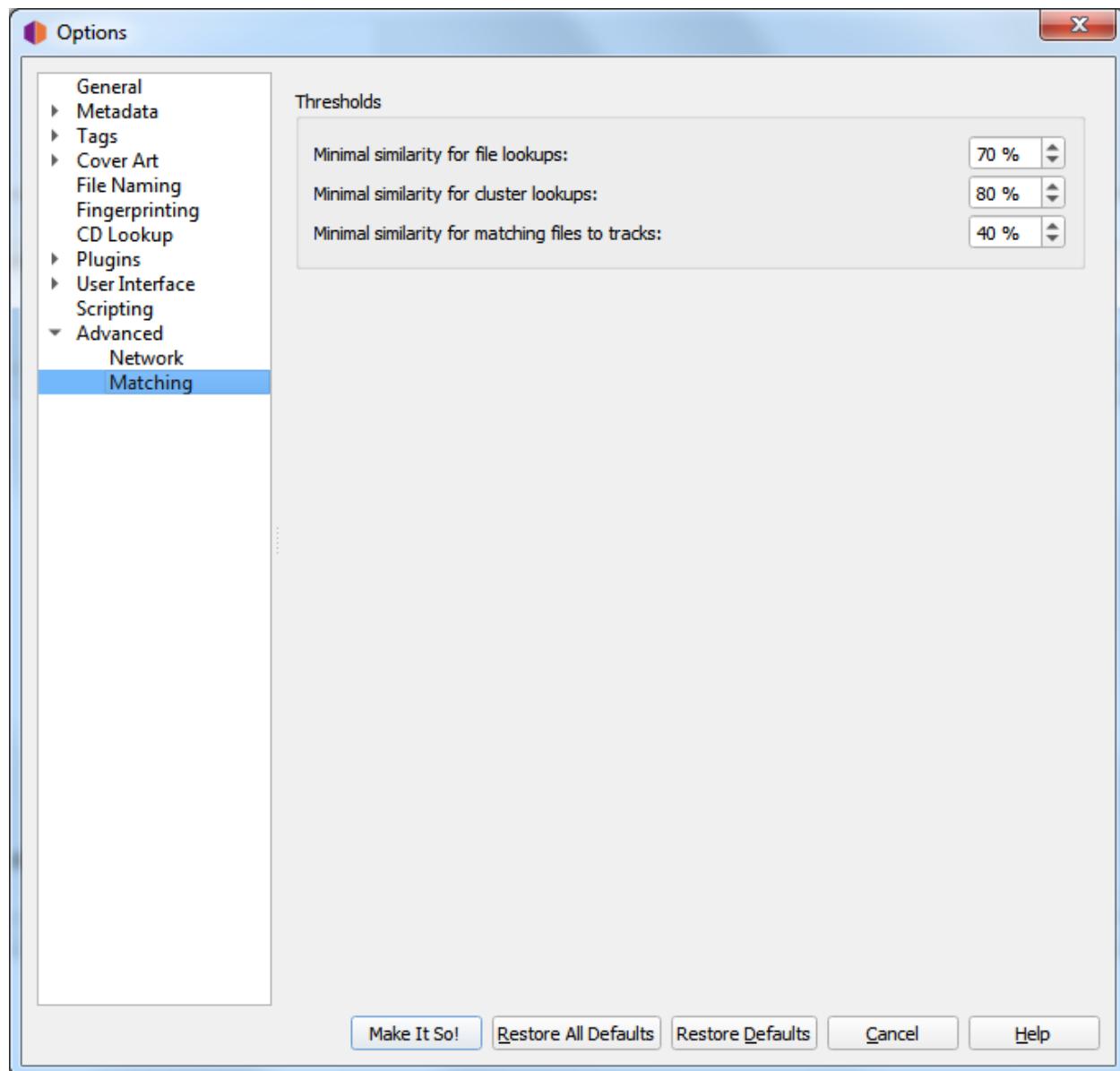
Cela identifie le port par défaut que Picard écoutera pour l'intégration du navigateur. Si le port n'est pas disponible, Picard essaiera d'augmenter le numéro de port de un jusqu'à ce qu'il trouve un port libre.

Écoutez uniquement sur localhost

Par défaut, Picard limitera l'accès au port d'intégration du navigateur à votre machine locale. La désactivation de cette option exposera le port sur votre réseau, vous permettant de demander à Picard de charger une version ou un enregistrement spécifique via le réseau. Par exemple, cela serait utilisé pour l'application [Picard Barcode Scanner](#) pour Android.

Avertissement: N'exposez le port en externe que lorsque vous en avez réellement besoin et uniquement sur les réseaux de confiance. L'exposition des ports d'application via le réseau peut ouvrir des failles de sécurité potentielles sur votre système.

Correspondance



Il est recommandé à la plupart des utilisateurs de laisser ces paramètres à leurs valeurs par défaut. Pour les utilisateurs avancés, ceux-ci vous permettent d'ajuster la façon dont Picard fait correspondre vos fichiers et vos clusters aux versions et pistes de MusicBrainz.

Similitude minimale pour les recherches de fichiers

Plus la valeur en pourcentage est élevée, plus les métadonnées d'un fichier individuel doivent être similaires aux métadonnées de MusicBrainz pour qu'elles correspondent à une version sur le côté droit.

Similitude minimale pour les recherches de cluster

Plus la valeur de pourcentage est élevée, plus un cluster de fichiers du

volet de gauche doit être similaire à une version de MusicBrainz pour que l'ensemble du cluster soit mis en correspondance avec une version sur le côté droit.

Similitude minimale pour faire correspondre les fichiers aux pistes

Plus la valeur en pourcentage est élevée, plus les métadonnées d'un fichier individuel doivent être similaires aux métadonnées de MusicBrainz pour qu'elles correspondent à une version sur le côté droit.

Si vous n'avez absolument aucune métadonnée dans vos fichiers actuels et que vous utilisez « Scan » pour faire correspondre les pistes, vous constaterez peut-être que vous devez réduire la valeur de « Similitude minimale pour faire correspondre les fichiers aux pistes » afin que Picard corresponde aux fichiers dans une version. Sinon, vous constaterez peut-être que Picard associe la piste à une version mais ne sait pas quelle piste est correcte; et le laisse dans un groupe « fichiers sans correspondance » dans cette version.

En règle générale, abaisser les pourcentages peut augmenter les chances de trouver une correspondance au risque de faux positifs et de correspondances incorrectes.

CHAPTER 7

Balises & Variables

Cela décrit à la fois les balises qui sont enregistrées dans vos fichiers musicaux et peuvent être lues par votre lecteur de musique, et les variables Picard qui peuvent être utilisées dans les scripts Picard pour le balisage, le changement de nom de fichier et dans plusieurs autres paramètres mineurs.

Toutes les balises sont également disponibles en tant que variables, mais les variables supplémentaires qui commencent par un trait de soulignement “_” ne sont pas enregistrées en tant que balises dans vos fichiers de musique (par exemple, `_my_tag_not_saved`).

Les variables sont utilisées dans les scripts en enveloppant le nom entre des caractères pour cent “%” (par exemple, `%title%`).

Certaines variables peuvent contenir plus d'une valeur (par exemple `musicbrainz_artistid`), et si vous souhaitez utiliser une seule des valeurs, vous devrez utiliser des fonctions de script spéciales pour y accéder ou les définir. Pour accéder à toutes les valeurs multiples à la fois, utilisez la variable normalement et Picard les combinera en une seule chaîne séparée par un point-virgule et un espace (par exemple: « Item 1; Item 2; Item 3 »).

Si une description de balise indique une version plus récente de Picard que la version officielle actuelle sur la page de téléchargement, alors la balise est une fonctionnalité bêta qui sera disponible dans la prochaine version officielle. Une description de la façon d'accéder à ces versions bêta pour les tests peut être trouvée sur la [page de téléchargement Picard](#) sur le site Web.

7.1 Balises de base

Les balises suivantes sont remplies à partir des données MusicBrainz pour la plupart des versions, sans aucun paramètre Picard spécial.

Tous ces éléments sont également disponibles en tant que variables à utiliser dans les scripts Picard (pour le balisage, pour le changement de nom de fichier et dans plusieurs autres paramètres plus mineurs) en les enveloppant entre des symboles de pourcentage "%" (par exemple, %title%).

Note: Les balises ne seront pas créées et ne seront pas disponibles en tant que variables si aucune valeur n'a été récupérée pour la balise dans la base de données MusicBrainz.

Note: Certaines de ces balises ne sont prises en charge que pour certains types de fichiers ou formats de balises. Veuillez consulter la section [Picard Tag Mapping](#) pour plus de détails.

acoustid_fingerprint

AcoustID Fingerprint pour la piste.

acoustid_id

AcoustID associé à la piste.

album

Titre du communiqué.

albumartist

Artiste ou artistes principalement crédités sur la sortie.

albumartistsort

Libérez le nom de tri de l'artiste (par exemple: « Beatles, The »).

albumsrt

Nom de tri du titre de publication.

artist

Suivre les noms des artistes. (chaîne)

artists

Suivre les noms des artistes. (multi-valeur) (*depuis Picard 1.3*)

artistsort

Nom de tri de l'artiste de piste.

asin

Numéro d'identification standard d'Amazon - le numéro identifiant l'article sur Amazon.

barcode

Le code-barres attribué à la version.

bpm

Battements par minute de la piste. Uniquement disponible pour le script de dénomination de fichier.

catalognumber

Les numéros attribués à la version par les étiquettes, qui se trouvent souvent sur le dos ou près du code-barres Il peut y en avoir plusieurs, en particulier lorsque plusieurs étiquettes sont impliquées.

comment

Le commentaire entré pour aider à distinguer une version d'une autre (par exemple: version Deluxe avec 2 pistes bonus).

compilation

(depuis Picard 1.3, compatible avec iTunes) 1 pour les albums Various Artist, sinon 0.

(Picard 1.2 ou précédent) 1 si plusieurs artistes de piste (y compris les artistes en vedette), sinon 0.

copyright

Contenir un message de copyright pour le détenteur du copyright du son original, commencer par un an et un espace.

date

Date de sortie (AAAA-MM-JJ) - la date à laquelle la version a été publiée.

director

Le réalisateur d'une piste vidéo tel que fourni par la relation Video Director dans MusicBrainz. (Depuis Picard 2.6)

discid

L'ID de disque est le numéro de code que MusicBrainz utilise pour lier un CD physique à une liste de versions. Ceci est basé sur les informations de table des matières (TOC) lues à partir du disque.

discnumber

Numéro du disque de cette version contenant cette piste.

discsubtitle

Le titre multimédia attribué à un disque spécifique.

encodedby

Encodé par (personne ou organisation). Uniquement disponible pour le script de dénomination de fichier.

encodersettings

Paramètres d'encodeur utilisés. Uniquement disponible pour le script de dénomination de fichier.

isrc

Un code standard international pour identifier de manière unique les enregistrements sonores et les enregistrements vidéo de musique. Voir [Wikipedia](#) pour plus d'informations. (*depuis Picard 0,12*)

key

Clé de la musique.

label

Libérez les noms d'étiquette.

language

Travaillez la langue des paroles selon [ISO 639-3](#) si les relations de piste sont activées dans Options et qu'un travail connexe existe. (*depuis Picard 0,10*)

lyrics

Paroles du titre.

media

Format de sortie (par exemple: CD). Consultez la page [Format de publication](#) sur le site Web MusicBrainz pour plus d'informations.

musicbrainz_albumartistid

Libérez l'identifiant MusicBrainz de l'artiste.

musicbrainz_albumid

L'identifiant MusicBrainz de la version.

musicbrainz_artistid

L'identifiant MusicBrainz de l'artiste de la piste.

musicbrainz_discid

L'ID du disque si l'album a été ajouté en utilisant « *Outils → Rechercher un CD* ». (*depuis Picard 0,12*)

musicbrainz_originalalbumid

L'identifiant MusicBrainz original de la version.

musicbrainz_originalartistid

L'identifiant MusicBrainz original de l'artiste du morceau.

musicbrainz_recordingid

L'identifiant MusicBrainz pour l'enregistrement.

musicbrainz_releasegroupid

Identifiant MusicBrainz pour le groupe de versions.

musicbrainz_releasetrackid

L'identifiant MusicBrainz pour la piste de version. (*depuis Picard 1.3*)

musicbrainz_trackid

Identificateur MusicBrainz de la piste.

musicbrainz_workid

L'identifiant MusicBrainz pour l'œuvre.

originalalbum

Le titre de la version la plus ancienne du groupe de publication destiné au titre de l'enregistrement original.

originalartist

L'artiste de la piste de la version la plus ancienne du groupe de publication destiné aux interprètes de l'enregistrement original.

originaldate

La date de sortie d'origine au format AAAA-MM-JJ. Par défaut, il s'agit de la version la plus ancienne du groupe de versions. Cela peut fournir, par exemple, la date de sortie de la version vinyle de ce que vous avez sur CD. (*Inclus en standard à partir de Picard 0.15, et en utilisant le plugin Original Release Date si vous utilisez toujours une version non-NGS antérieure à Picard 0.15*)

Note: Il s'agit des mêmes informations fournies dans la variable `_releasegroup_firstrleasedate`, et elles sont cohérentes sur toutes les pistes de la version. Si vous préférez que cette balise soit renseignée avec la date du premier enregistrement de la piste dans la base de données, qui sera probablement différente pour chaque piste de la version, cela peut être réalisé en activant un script de marquage sur une ligne comme `$set(originaldate,%_recording_firstrleasedate%)`. Sachez que définir cela peut entraîner la dispersion d'une version dans plusieurs répertoires si vous utilisez `%originaldate%` dans le cadre de la partie chemin de votre script de nommage de fichier.

Note: Si vous stockez des balises dans des fichiers MP3 en tant que ID3v2.3 (qui est la version compatible Windows et iTunes), la date d'origine ne peut

être stockée que comme une année.

originalyear

L'année de la date de sortie d'origine au format AAAA. Par défaut, il s'agit de la version la plus ancienne du groupe de versions. Cela peut fournir, par exemple, l'année de sortie de la version vinyle de ce que vous avez sur CD.

releasecountry

Le pays dans lequel la libération a été publiée.

releasestatus

Le statut de la version indiquant le statut « officiel » de la version. Les valeurs possibles incluent officiel, promotionnel, bootleg et pseudo-release.

releasetype

Le type de groupe de version (voir aussi [*_primaryreleasetype*](#) et [*_secondaryreleasetype*](#)).

script

Le script utilisé pour écrire la liste des morceaux de la sortie. Les valeurs possibles sont tirées du standard [ISO 15924](#). (depuis Picard 0,10)

subtitle

Utilisé pour les informations directement liées au titre du contenu.

title

Le titre de la piste.

titlesort

Le nom de tri du titre de la piste.

totaldiscs

Le nombre total de disques dans cette version

totaltracks

Le nombre total de pistes sur ce disque.

tracknumber

Le numéro de la piste sur ce disque.

website

Utilisé pour le site officiel de l'artiste.

7.2 Balises avancées

Vous pouvez rendre des balises supplémentaires disponibles en activant les paramètres *Relations avancées* et *Utiliser les genres de MusicBrainz* paramètres dans Picard.

Note: Les balises ne seront pas créées et ne seront pas disponibles en tant que variables si aucune valeur n'a été récupérée pour la balise dans la base de données MusicBrainz.

Note: Certaines de ces balises ne sont prises en charge que pour certains types de fichiers ou formats de balises. Veuillez consulter la section *Picard Tag Mapping* pour plus de détails.

7.2.1 Balises de relation avancées

Si vous activez le balisage avec les relations avancées, vous obtenez ces balises supplémentaires:

arranger

Type de relation arrangeur (sorties, enregistrements, œuvres), type de relation instrumentateur, type de relation orchestrateur (*depuis Picard 0.10*)

composer

Type de relation de compositeur

composersort

Nom de tri du type de relation de compositeur

conductor

Type de relation de chef d'orchestre (sorties, enregistrements), type de relation de maître de chœur (sorties, enregistrements)

djmixer

Type de relation Mix-DJ (*depuis Picard 0.9*)

engineer

Type de relation ingénieur

license

Type de relation de licence (sorties, enregistrements) (*depuis Picard 1.0*)

lyricist

Type de relation parolier

mixer

Type de relation ingénieur (« Mixé par ») (*depuis Picard 0.9*)

performer:<type>

Type de relation interprète (sorties - voix / instruments, enregistrements - voix / instruments), <type> peut être « vocal », « guest guitar », « solo violin »,...

Type de relation avec l'orchestre (sorties, enregistrements), <type> est « orchestra »

Type de relation Maître de concert (sorties, enregistrements), <type> est « concertmaster »

producer

Type de relation producteur

remixer

Type de relation de remixeur

work

Nom du travail (*depuis Picard 1.3*)

writer

Type de relation écrivain (*depuis Picard 1.0*). Non écrit automatiquement dans la plupart des formats de fichiers. Vous pouvez fusionner cela avec des compositeurs avec un script comme:

```
$copymerge(composer, writer)
```

7.2.2 Balises de genre

Si vous activez Utiliser les genres de MusicBrainz, vous obtenez:

genre

Informations sur le genre de MusicBrainz (*depuis Picard 2.1, les versions antérieures utilisaient des balises folksonomy*)

7.3 Variables de base

Ces variables sont renseignées à partir des données MusicBrainz pour la plupart des versions, sans aucun paramètre Picard particulier.

Note: Les variables ne seront pas créées si aucune valeur n'a été récupérée pour la variable à partir de la base de données MusicBrainz.

_absolutetracknumber

Le numéro absolu de cette piste sans tenir compte du numéro de disque (c'est-à-dire: %_absolutetracknumber% de %_totalbumtracks%). Par exemple, cette valeur serait 11 pour la deuxième piste du disque 2 où le disque 1 a 9 pistes. (*depuis Picard 1.3*)

_albumartists

Les noms des artistes de l'album (valeurs multiples). (*depuis Picard 1.3*)

_albumartists_sort

Les noms de tri des artistes de l'album (valeurs multiples). (*depuis Picard 1.3*)

_artists_sort

Les noms de tri des artistes de la piste (valeurs multiples). (*depuis Picard 1.3*)

_datatrack

Réglé sur 1 si la piste est une « piste de données ». (*depuis Picard 1.3.1*)

_discpregap

Réglé sur 1 si le disque sur lequel se trouve la piste a une « piste de pré-insertion ». (*depuis Picard 1.4*)

_multiartist

0 si toutes les pistes de l'album ont le même artiste principal, 1 sinon. (*depuis Picard 1.3*)

_pregap

Réglé sur 1 si la piste est une « piste pré-assemblée ». (*depuis Picard 1.3.1*)

_primaryreleasetype

Le type de groupe principal de version (ex: album, single, ep, diffusion ou autre).

_rating

Note 0-5 par les utilisateurs de MusicBrainz.

_recording_firstreleasedate

Date du premier enregistrement d'une piste au format AAAA-MM-JJ. (*Depuis Picard 2.6*)

_releaseannotation

Le commentaire d'annotation pour la version. (*depuis Picard 2.6*)

_releasecomment

Le commentaire d'homonymie pour la version. (*depuis Picard 0.15*)

_releasecountries

Ceci fournit la liste complète des pays de la version pour la version comme une variable à plusieurs valeurs. (*depuis Picard 2.3.1*)

_releasegroup

Le titre du groupe de versions qui est généralement le même que le titre de l'album, mais peut être différent.

_releasegroupcomment

Le commentaire d'homonymie pour le groupe de version.

_releasegroup_firstreleasedate

La date de la version la plus ancienne du groupe de versions au format AAAA-MM-JJ. Ceci est destiné à fournir, par exemple, la date de sortie de la version vinyle de ce que vous avez sur CD. (*Depuis Picard 2.6*)

Note: Ce sont les mêmes informations fournies par défaut dans la balise `originaldate`.

_releaselanguage

La langue de la version selon [ISO 639-3](#). (*depuis Picard 0.10*)

_secondaryreleasetype

Zéro ou plus de types secondaires pour le groupe de versions (c'est-à-dire: livre audio, compilation, DJ-mix, interview, live, mixtape / street, remix, bande sonore ou speechword).

_totalalbumtracks

Le nombre total de pistes sur tous les disques de cette version.

7.4 Variables de fichier

Ces variables sont renseignées à partir des données MusicBrainz pour la plupart des versions, sans aucun paramètre Picard particulier.

Note: Les variables qui reposent sur les informations des fichiers (par exemple: `_bitrate`) ne sont disponibles que pour une utilisation sur les pistes avec des fichiers joints, lors de l'exécution manuelle de scripts sur des fichiers ou dans le script de dénomination de fichier.

Avertissement: Avant la version 2.5, Picard ne prenait pas en charge l'utilisation de variables de fichier dans les scripts de balisage.

`_bitrate`

Débit binaire approximatif en kbps.

`_bits_per_sample`

Bits de données par échantillon.

`_channels`

Nombre de canaux audio dans le fichier.

`_dirname`

Le nom du répertoire contenant le fichier au moment de son chargement dans Picard. (*depuis Picard 1.1*)

`_extension`

L'extension du fichier. (*depuis Picard 0.9*)

`_filename`

Le nom du fichier sans extension. (*depuis Picard 1.1*)

`_format`

Le format multimédia du fichier (par exemple: MPEG-1 Audio).

`_length`

La longueur de la piste au format mins:secs.

`_sample_rate`

Le nombre d'échantillons de numérisation par seconde (Hz).

7.5 Variables avancées

Si vous activez le balisage avec *Relations avancées*, vous obtenez ces variables supplémentaires:

Note: Les variables ne seront pas créées si aucune valeur n'a été récupérée pour la variable à partir de la base de données MusicBrainz.

_performance_attributes

Liste des attributs de performance de l'œuvre (ex: « live », « cover », « medley »). Utilisez `$inmulti` pour rechercher un type spécifique (c'est-à-dire: `$if($inmulti(%_performance_attributes%,medley), (Medley),)`). (depuis Picard 1.3)

_recordingcomment

Enregistrement d'un commentaire d'homonymie. (depuis Picard 0,15)

_recordingtitle

Titre de l'enregistrement - normalement le même que le titre de la piste, mais peut être différent.

7.6 Balises de musique classique

Avec l'aide de plugins comme « Classical Extras » ou « Work & Movement », vous pouvez utiliser les balises suivantes pour marquer votre musique classique.

movement

Nom du mouvement (ex: « Andante con moto »).

movementnumber

Numéro de mouvement en chiffres arabes (par exemple: « 2 »). Les joueurs prenant explicitement en charge cette balise l'afficheront souvent en chiffres romains (par exemple: « II »).

movementtotal

Nombre total de mouvements dans l'œuvre (par exemple: « 4 »).

showmovement

Afficher le travail et le mouvement: si cette balise est définie sur « 1 », les lecteurs prenant en charge cette balise, comme iTunes et MusicBee, afficheront l'œuvre, le numéro de mouvement et le nom du mouvement au lieu du titre de la piste. Par exemple, la piste sera affichée comme « Symphonie n ° 5 en ut mineur, op. 67: II. Andante con moto » quelle que soit la valeur de la balise de titre.

work

Le nom de l'œuvre dans son ensemble (par exemple: « Symphonie n ° 5 en ut mineur, op. 67 »).

Note: Si vous utilisez iTunes avec des fichiers MP3, vous devez activer l'option « Enregistrer le regroupement et le travail compatibles avec iTunes » pour que le travail s'affiche correctement.

7.7 Balises des plugins

Plugins de Picard [Les plugins](#) peuvent ajouter plus de balises. Voici quelques exemples.

7.7.1 Plugin Last.fm

genre

Pseudo-genre basé sur des balises de folksonomie.

7.7.2 Plugin de variables d'artistes supplémentaires

Album Variables

_artists_album_primary_id

L'ID de l'artiste principal / premier album répertorié

_artists_album_primary_std

L'artiste principal / premier album répertorié (standardisé)

_artists_album_primary_cred

L'artiste principal / premier album répertorié (comme crédité)

_artists_album_primary_sort

L'artiste principal / premier album répertorié (nom de tri)

_artists_album_additional_id

Les identifiants de tous les artistes de l'album répertoriés, à l'exception de l'artiste principal / premier, sous forme de valeurs multiples

_artists_album_additional_std

Tous les artistes de l'album répertoriés (standardisés) à l'exception de l'artiste principal / premier, séparés par des chaînes fournies à partir de l'entrée de version

_artists_album_additional_cred

Tous les artistes de l'album répertoriés (comme crédités) à l'exception de l'artiste principal / premier, séparés par des chaînes fournies à partir de l'entrée de version

_artists_album_additional_sort

Tous les artistes de l'album répertoriés (trier les noms) à l'exception de l'artiste principal / premier, séparés par des chaînes fournies à partir de l'entrée de version

_artists_album_additional_std_multi

Tous les artistes de l'album répertoriés (standardisés) à l'exception du premier / du premier artiste, en tant que valeur multiple

_artists_album_additional_cred_multi

Tous les artistes de l'album répertoriés (comme crédités), à l'exception de l'artiste principal / premier, en tant que valeur multiple

_artists_album_all_std

Tous les artistes de l'album répertoriés (standardisés), séparés par des chaînes fournies à partir de l'entrée de version

_artists_album_all_cred

Tous les artistes de l'album répertoriés (comme crédités), séparés par des chaînes fournies à partir de l'entrée de version

_artists_album_all_sort

Tous les artistes de l'album répertoriés (noms de tri), séparés par des chaînes fournies à partir de l'entrée de version

_artists_album_all_std_multi

Tous les artistes d'album répertoriés (standardisés), en tant que valeur multiple

_artists_album_all_cred_multi

Tous les artistes de l'album répertoriés (comme crédités), comme valeur multiple

_artists_album_all_sort_primary

L'artiste principal / premier album répertorié (nom de tri) suivi de tous les artistes d'album supplémentaires (standardisés), séparés par des chaînes fournies à partir de l'entrée de version

_artists_album_all_count

Le nombre d'artistes répertoriés comme artistes d'album

Track Variables

_artists_track_primary_id

L'ID de l'artiste principal / premier morceau répertorié

_artists_track_primary_std

L'artiste principal / premier morceau répertorié (standardisé)

_artists_track_primary_cred

L'artiste principal / premier morceau répertorié (comme crédité)

_artists_track_primary_sort

L'artiste principal / premier morceau répertorié (nom de tri)

_artists_track_additional_id

Les identifiants de tous les artistes de piste répertoriés à l'exception de l'artiste principal / premier, sous forme de valeurs multiples

_artists_track_additional_std

Tous les artistes de piste répertoriés (standardisés) à l'exception de l'artiste principal / premier, séparés par des chaînes fournies à partir de l'entrée de piste

_artists_track_additional_cred

Tous les artistes de piste répertoriés (comme crédités) à l'exception de l'artiste principal / premier, séparés par des chaînes fournies à partir de l'entrée de piste

_artists_track_additional_sort

Tous les artistes de piste répertoriés (trier les noms) à l'exception de l'artiste principal / premier, séparés par des chaînes fournies à partir de l'entrée de piste

_artists_track_additional_std_multi

Tous les artistes de piste répertoriés (standardisés) à l'exception de l'artiste principal / premier, en tant que valeur multiple

_artists_track_additional_cred_multi

Tous les artistes de piste répertoriés (comme crédités) à l'exception du premier / principal artiste, en tant que valeur multiple

_artists_track_all_std

Tous les artistes de piste répertoriés (standardisés), séparés par des chaînes fournies à partir de l'entrée de piste

_artists_track_all_cred

Tous les artistes de piste répertoriés (comme crédités), séparés par des chaînes fournies à partir de l'entrée de piste

_artists_track_all_sort

Tous les artistes de piste répertoriés (nom de tri), séparés par des chaînes fournies à partir de l'entrée de piste

_artists_track_all_std_multi

Tous les artistes de piste répertoriés (standardisés), en tant que valeur multiple

_artists_track_all_cred_multi

Tous les artistes de piste répertoriés (comme crédités), en tant que valeur multiple

_artists_track_all_sort_primary

L'artiste principal / premier de la piste répertorié (nom de tri) suivi de tous les artistes de piste supplémentaires (standardisés), séparés par des chaînes fournies à partir de l'entrée de piste

_artists_track_all_count

Le nombre d'artistes répertoriés comme artistes de piste

Note: Certains plugins effectuent un grand nombre d'appels au service Web pour obtenir des données supplémentaires spécifiques à la piste, telles que l'interprète et les relations de travail, de sorte que le chargement d'un grand nombre d'albums et de pistes peut prendre beaucoup de temps. Le problème de temps peut être partiellement résolu en exploitant un serveur MusicBrainz local avec la limitation de débit désactivée. Veuillez consulter le projet [MusicBrainz Server](#) sur GitHub pour plus d'informations.

7.8 Autres informations

Pour des détails techniques sur l'écriture des balises dans les fichiers, consultez la section [Picard Tag Mapping](#).

Si vous définissez des variables qui ne sont pas connues de Picard, celles-ci seront enregistrées en tant que nouvelles balises dans les fichiers basés sur ID3, MP4, APEv2 et Vorbis. Ils ne seront pas enregistrés dans des fichiers ASF.

- Pour les fichiers basés sur ID3, ces balises seront enregistrées et rechargées à partir de cadres d'informations textuelles définies par l'utilisateur ID3 (TXXX).
- Pour les fichiers MP4, ces balises seront enregistrées avec le préfixe `----:com.apple.iTunes:.` Ceci est largement compris par d'autres outils à utiliser pour les balises personnalisées.
- Pour les fichiers Vorbis et APEv2, ces balises seront enregistrées telles quelles.

Pour les balises basées sur ID3 (par exemple: fichiers MP3), vous pouvez également définir des balises ID3 directement à partir de vos scripts en définissant une variable spéciale commençant par `_id3:.`, par ex. `%_id3:TXXX:ma balise%`. Actuellement, ces

balises ne sont pas chargées dans les variables lorsque vous rechargez le fichier dans Picard (*depuis Picard 0.9*).

Note: L'enregistrement de balises personnalisées dans des fichiers MP4 est pris en charge depuis Picard 2.3. Les versions antérieures n'enregistreront ni ne chargeront les balises personnalisées dans les fichiers MP4.

CHAPTER 8

Script

Les scripts sont utilisés pour contrôler certains aspects du fonctionnement de Picard.

Il existe deux types de scripts utilisés dans Picard: le script de dénomination de fichier et les scripts de balisage. Ceux-ci sont gérés à partir des sections « Nom du fichier » et « Script » du menu « *Options → Options...* ».

Les scripts sont souvent abordés dans le [MetaBrainz Community Forum](#), et il existe un fil de discussion spécifique à [la dénomination de fichier et les extraits de script](#).

Voir aussi:

Veuillez vous référer à la section sur [*Scripts*](#) dans [*Extending Picard*](#) pour plus de détails sur les deux types de scripts, y compris comment et quand chacun des scripts est exécuté.

8.1 Syntaxe

La syntaxe est dérivée du [format de titre de Foobar2000](#). Il existe trois éléments de base: le texte, la variable et la fonction. Les variables sont constituées de caractères alphanumériques entourés de signes de pourcentage (par exemple: %artiste%). Les fonctions commencent par un signe dollar et se terminent par une liste d'arguments entre parenthèses (par exemple: \$lower(...)).

Note: Lorsque vous entrez des chaînes d'entrée dans des scripts Picard, vous devez échapper une barre oblique inverse « \ », un signe dollar « \$ », une virgule « , » et les parenthèses gauche et droite « (» et «) » afin de forcer Picard à ne pas les interpréter dans le cadre de la commande de script. Cela se fait en insérant une barre oblique inverse avant le caractère à échapper. Par exemple, pour définir une valeur de balise

sur (\$1,000,000), il faudrait la saisir sous la forme \$set(test_tag,\(\$1\,000\,000\)).

8.2 Variables de métadonnées

Voir [Balises & Variables](#) pour la liste des variables fournies par Picard.

Les variables de Picard peuvent être soit de simples variables contenant une seule chaîne de texte, soit des variables à valeurs multiples contenant plusieurs chaînes de texte. Dans les scripts, les variables à valeurs multiples sont automatiquement converties en une seule chaîne de texte en joignant les valeurs par un point-virgule “;”, sauf lorsqu’elles sont utilisées avec des fonctions spéciales à valeurs multiples.

Note: La liste complète des fonctions de script disponibles est couverte dans le chapitre suivant.

CHAPTER 9

Fonctions de script

Voici une liste des fonctions de script Picard regroupées par type de fonction.

9.1 Fonctions d'affectation

Ces fonctions sont utilisées pour affecter (ou annuler l'attribution) d'une valeur à une balise ou une variable. Les fonctions de script d'affectation sont:

9.1.1 \$copy

Utilisation: **\$copy(target,source)**

Catégorie: affectation

Mis en œuvre: Picard 0.9

La description:

Copie les métadonnées de la variable source vers target. La différence avec `$set(target,%source%)` est que `$copy(target, source)` copie les variables à valeurs multiples sans les aplatis.

Notez que si la variable target existe déjà, elle sera écrasée par source.

Exemple:

Les instructions suivantes donneront les valeurs de target comme indiqué:

```
$set(source,)  
$set(target,This will be overwritten)  
$copy(target,source)      ==>  ""  
  
$set(source,one)  
$copy(target,source)      ==>  "one"  
  
$setmulti(source,one)  
$copy(target,source)      ==>  "one"  
  
$setmulti(source,one; two)  
$copy(target,source)      ==>  "one; two"
```

9.1.2 \$copymerge

Utilisation: **\$copymerge(target,source)**

Catégorie: affectation

Mis en œuvre: Picard 1.0

La description:

Fusionne les métadonnées de la variable source dans target, en supprimant les doublons et en les ajoutant à la fin, conservant ainsi l'ordre d'origine. Comme [\\$copy](#), cela copiera également les variables à valeurs multiples sans les aplatisir. Après l'opération, target sera une variable à valeurs multiples.

Notez que les noms de variable pour target et source sont passés directement sans les entourer de signes de pourcentage "%".

Exemple:

Les instructions suivantes donneront les valeurs de target comme indiqué:

```
$set(target,)  
$set(source,one)  
$copymerge(target,source)      ==>  "one"  
  
$set(target,zero)  
$set(source,one)  
$copymerge(target,source)      ==>  "zero; one"  
  
$set(target,zero)  
$setmulti(source,one; two)  
$copymerge(target,source)      ==>  "zero; one; two"  
  
$setmulti(target,zero; two)  
$setmulti(source,one; two)  
$copymerge(target,source)      ==>  "zero; two; one"
```

(suite sur la page suivante)

(suite de la page précédente)

```
$set(target,zero; one; zero)
$set(source,one; two; three)
$copymerge(target,source)    ==> "zero, one; two; three"
```

9.1.3 \$delete

Utilisation: **\$delete(nom)**

Catégorie: affectation

Mis en œuvre: Picard 2.1

La description:

Annule la variable nom et marque la balise à supprimer.

Ceci est similaire à \$unset(nom) mais marque également la balise à supprimer. Par exemple, exécuter \$delete(genre) supprimera en fait la balise « genre » d'un fichier lors de l'enregistrement.

Exemple:

Les instructions suivantes exécuteront les actions indiquées:

```
$delete(genre) ==> Remove the "genre" tag from a file when saving
```

9.1.4 \$set

Utilisation: **\$set(nom,valeur)**

Catégorie: affectation

La description:

Définit la variable nom sur valeur. La valeur d'une variable est disponible pour d'autres fonctions de script si elle est comprise entre les caractères "%" (par exemple: %name%). Si nom est une autre variable (par exemple: %indirect%), la valeur de la variable sera utilisée comme nom. Cela permet la création de variables nommées dynamiquement.

Note: Pour créer une variable qui peut être utilisée pour la chaîne de dénomination du fichier, mais qui ne sera pas écrite comme une balise dans le fichier, préfixez le nom de la variable avec un trait de soulignement. %something% créera une balise something; %_something% ne le sera pas.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(comment,Testing) ==> "Testing" will be written to the "comment" tag  
$set(_hidden,Testing) ==> "_hidden" variable will not be written  
  
$set(_base,redirect)  
$set(%_base%,Testing) ==> "Testing" will be written to the "redirect" tag
```

9.1.5 \$setmulti

Utilisation: **\$setmulti(nom,valeur[, séparateur])**

Catégorie: affectation

Mis en œuvre: Picard 1.0

La description:

Définit la variable nom sur valeur, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour forcer la valeur à revenir dans un multi approprié -variable valorisée. Cela peut être utilisé pour opérer sur des variables à valeurs multiples sous forme de chaîne, puis les remettre en tant que variable à valeurs multiples appropriée.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$setmulti(genre,$lower(%genre%)) ==> all "genre" elements in lower case  
$setmulti(alpha,A; B; C) ==> 3 elements ("A", "B" and "C")  
$setmulti(mixed,A:A; B:B,:) ==> 3 elements ("A", "A; B" and "B")
```

9.1.6 \$unset

Utilisation: **\$unset(nom)**

Catégorie: affectation

La description:

Annule la variable nom. La fonction permet aux caractères génériques d'annuler certaines balises (fonctionne avec “performer:”, “comment:”, and “lyrics:”).

Exemple:

Ce qui suit annule toutes les balises d'intervenant:

```
$unset(performer:*)
```

9.2 Fonctions de texte

Ces fonctions permettent de gérer du texte (ex: extraire, remplacer ou formater) dans des balises ou des variables. Les fonctions de script de texte sont:

9.2.1 \$delprefix

Utilisation: **\$delprefix(texte[,préfixes])**

Catégorie: texte

Mis en œuvre: Picard 1.3

La description:

Supprime les préfixes spécifiés du début de texte. N'importe quel nombre de préfixes peut être spécifié, séparés par des virgules. Si aucun préfixe n'est spécifié, « A » et « The » sont utilisés par défaut. Notez que la correspondance est sensible à la casse.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$delprefix(The Beatles)</code>	<code>==> "Beatles"</code>
<code>\$delprefix(The Beatles,)</code>	<code>==> "The Beatles"</code>
<code>\$delprefix(THE Beatles)</code>	<code>==> "THE Beatles"</code>
<code>\$delprefix(THE Beatles,THE)</code>	<code>==> "Beatles"</code>
<code>\$delprefix(The Beatles,A,An)</code>	<code>==> "The Beatles"</code>

9.2.2 \$find

Utilisation: **\$find(meule de foin,aiguille)**

Catégorie: texte

Mis en œuvre: Picard 2.3

La description:

Renvoie l'index de base zéro de la première occurrence de aiguille dans meule de foin, ou une chaîne vide si aiguille n'a pas été trouvée. Les comparaisons sont sensibles à la casse. Si aiguille est vide, il correspondra au début de meule de foin et renverra « 0 ». La fonction ne prend pas en charge les caractères génériques.

Note: Avant Picard 2.3.2, \$find retournait « -1 » si aiguille n'était pas trouvée.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$find(abcdef,a)      ==> "0"  
$find(abcdef,c)      ==> "2"  
$find(abcdef,cd)     ==> "2"  
$find(abcdef,g)      ==> ""  
$find(abcdef,B)      ==> ""  
$find(,a)             ==> ""  
$find(abcdef,)        ==> "1"
```

9.2.3 \$firstalphachar

Utilisation: **\$firstalphachar(texte[,nonalpha])**

Catégorie: texte

Mis en œuvre: Picard 0.12

La description:

Renvoie le premier caractère de texte en majuscules. Si texte ne commence pas par un caractère alphabétique, alors nonalpha est renvoyé à la place. Si nonalpha n'est pas spécifié, la valeur par défaut « # » sera utilisée.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$firstalphachar(abc)          ==> "A"  
$firstalphachar(123)          ==> "#"  
$firstalphachar(***)         ==> "#"  
$firstalphachar(***,)        ==> ""  
$firstalphachar(***,^)       ==> "^"  
$firstalphachar(***,non-alpha) ==> "non-alpha"
```

9.2.4 \$firstwords

Utilisation: **\$firstwords(texte,longueur)**

Catégorie: texte

Mis en œuvre: Picard 0.12

La description:

Tronquez texte en longueur, ne renvoyant que les mots complets de texte qui tiennent dans les caractères de longueur. Si longueur est inférieur à 0, alors la valeur utilisée est le nombre de caractères dans texte plus longueur (par exemple: \$firstwords(un deux trois, -3) est identique à \$firstwords(un deux trois, 10) ''). Si ``longueur est manquant ou un nombre négatif supérieur au nombre de caractères dans texte, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$firstwords(Once upon a time,)      ==> ""
$firstwords(Once upon a time,0)    ==> ""
$firstwords(Once upon a time,3)    ==> ""
$firstwords(Once upon a time,7)    ==> "Once"
$firstwords(Once upon a time,-3)   ==> "Once upon a"
$firstwords(Once upon a time,-30)  ==> ""
```

9.2.5 \$get

Utilisation: **\$get(nom)**

Catégorie: texte

La description:

Renvoie la variable nom (équivalent à % nom%) ou une chaîne vide si nom n'a pas été défini. Si nom est une autre variable (par exemple %indirect%), la valeur de la variable sera utilisée comme nom. Cela permet la récupération de variables nommées dynamiquement.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,This is foo)
$get(bar,foo)
$get(foo)           ==> "This is foo"
$get(bar)           ==> "foo"
$get(%bar%)        ==> "This is foo"
$get(baz)           ==> "" ('baz' has not been set to a value)
```

9.2.6 \$initials

Utilisation: **\$initials(texte)**

Catégorie: texte

Mis en œuvre: Picard 0.12

La description:

Renvoie le premier caractère de chaque mot dans texte, s'il s'agit d'un caractère alphabétique.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,This is a test)
$initials(%foo%)           ==> "Tiat"
$initials(This is a test)   ==> "Tiat"
$initials(This is a 123 test) ==> "Tiat"
```

9.2.7 \$left

Utilisation: **\$left(texte,nombre)**

Catégorie: texte

La description:

Renvoie les premiers caractères nombre de texte. Si nombre est inférieur à 0, alors la valeur utilisée est le nombre de caractères dans texte plus nombre (par exemple: \$right(abcd,-1) est le même comme \$right(abcd,3)). Si nombre est manquant ou un nombre négatif supérieur au nombre de caractères dans texte, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$left(,)      ==> ""
$left(ABC,)    ==> ""
$left(ABC,0)    ==> ""
$left(ABC,2)    ==> "AB"
$left(ABC,4)    ==> "ABC"
$left(ABC,-2)   ==> "A"
$left(ABC,-4)   ==> ""
```

9.2.8 \$len

Utilisation: **\$len(texte)**

Catégorie: texte

La description:

Renvoie le nombre de caractères dans texte.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,)
$len(%foo%)    ==> "0"

$set(foo,ABC)
```

(suite sur la page suivante)

(suite de la page précédente)

```
$len(%foo%)    ==> "3"
$len()         ==> "0"
$len(ABC)      ==> "3"
```

9.2.9 \$lower

Utilisation: **\$lower(texte)**

Catégorie: texte

Mis en œuvre: Picard

La description:

Renvoie texte en minuscules.

Exemple:

L'instruction suivante renverra la valeur indiquée:

```
$title(tHe houR is upOn uS) ==> "the hour is upon us"
```

9.2.10 \$num

Utilisation: **\$num(nombre,longueur)**

Catégorie: texte

La description:

Renvoie nombre formaté en chiffres de longueur, où nombre et longueur sont des entiers et longueur ne peut pas être supérieur à 20.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$num(,)        ==> ""
$num(,1)       ==> "0"
$num(a,)       ==> ""
$num(a,5)      ==> "00000"
$num(123,5)    ==> "00123"
$num(1.23,5)   ==> "00000"
$num(123,)     ==> ""
$num(123,0)    ==> "123"
$num(123,1)    ==> "123"
$num(123,20)   ==> "00000000000000000000123"
$num(123,50)   ==> "00000000000000000000000000000000123"
```

(suite sur la page suivante)

(suite de la page précédente)

```
$num(123,5.5) ==> ""
$num(1.23,10) ==> "0000000000"
```

9.2.11 \$pad

Utilisation: **\$pad(texte,longueur,caractère)**

Catégorie: texte

La description:

Remplit le texte à la longueur fournie en ajoutant autant de copies de caractère que nécessaire au début de la chaîne. Pour que la longueur rempile soit correcte, caractère doit avoir exactement un caractère de longueur. Si longueur est inférieur au nombre de caractères dans texte, la fonction renverra texte. Si length est manquant ou n'est pas un nombre, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$pad(abc,5,+)
$pad(abc,0,+)
$pad(abc,5, )
$pad(abc,5,XY)
$pad(abc,,+)
$pad(abc,x,+)
```

==> "++abc"
==> "abc"
==> "abc"
==> "XYXYabc" (note final length is incorrect)
==> ""
==> ""

9.2.12 \$replace

Utilisation: **\$replace(texte,recherche,remplacement)**

Catégorie: texte

La description:

Remplace les occurrences de recherche dans texte par remplacement et renvoie la chaîne résultante.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,I like cats the best)
$replace(%foo%,cat,dog)
==> "I like dogs the best"

$set(foo,I like cats the best)
$set(bar,cat)
```

(suite sur la page suivante)

(suite de la page précédente)

<code>\$replace(%foo%,%bar%,dog)</code>	<code>==> "I like dogs the best"</code>
<code>\$set(foo,I like cats the best)</code>	
<code>\$set(bar,cat)</code>	
<code>\$set(baz,dog)</code>	
<code>\$replace(%foo%,%bar%,%baz%)</code>	<code>==> "I like dogs the best"</code>
<code>\$replace(I like cats the best,cat,dog)</code>	<code>==> "I like dogs the best"</code>
<code>\$replace(I like cats the best,pig,dog)</code>	<code>==> "I like cats the best"</code>
<code>\$replace(I like cats the best,cat,)</code>	<code>==> "I like s the best"</code>
<code>\$replace(Bad replace,,_)</code>	<code>==> "_B_a_d_ _r_e_p_l_a_c_e_"</code>

9.2.13 \$reverse

Utilisation: **\$reverse(texte)**

Catégorie: texte

La description:

Renvoie texte dans l'ordre inverse.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$set(foo,abcde)</code>	
<code>\$reverse(%foo%)</code>	<code>==> "edcba"</code>
<code>\$reverse(abcde)</code>	<code>==> "edcba"</code>

9.2.14 \$right

Utilisation: **\$right(texte,nombre)**

Catégorie: texte

La description:

Renvoie les derniers caractères nombre de texte. Si nombre est inférieur à 1, alors la valeur utilisée est le nombre de caractères dans texte plus nombre (par exemple: `$right(abcd,0)` est identique à `$right(abcd,4)`). Si nombre est manquant ou un nombre négatif supérieur au nombre de caractères dans texte, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$right(abcd,2)    ==> "cd"
$right(abcd,0)    ==> "cd"
$right(abcd,-1)   ==> "bcd"
$right(abcd,)     ==> ""
$right(abcd,-5)   ==> ""
```

9.2.15 \$rreplace

Utilisation: **\$rreplace(texte,motif,replacement)**

Catégorie: texte

La description:

Expression régulière remplacer. Cette fonction remplacera le groupe correspondant spécifié par motif par replacement dans texte. Pour plus d'informations sur les expressions régulières, veuillez consulter l'article sur [Wikipedia](#).

Note: Lors de la saisie d'expressions régulières dans des scripts Picard, vous devez échapper une barre oblique inverse « \ », un signe dollar « \$ », une virgule « , » et les parenthèses gauche et droite « (» et «) » pour forcer Picard à ne pas les interpréter dans le cadre de la commande de script. Cela se fait en insérant une barre oblique inverse avant le caractère à échapper. Par exemple, l'expression régulière ^\s*([0-9,.]*\$ devrait être saisie sous la forme ^\\s*\\([0-9\\,.]*\\)\$.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$rreplace(test \\(disc 1\\),\\s\\\\(disc \\d+\\\\\\),) ==> "test"
$rreplace(test,[t,)           ==> "test"
```

9.2.16 \$rsearch

Utilisation: **\$rsearch(texte,motif)**

Catégorie: texte

La description:

Recherche d'expression régulière. Cette fonction renverra le premier groupe correspondant spécifié par motif à partir de texte. Pour plus d'informations sur les expressions régulières, veuillez consulter l'article sur [Wikipedia](#).

Si une sous-expression marquée est définie à l'aide de parenthèses dans le modèle de recherche, seul le modèle capturé par la sous-expression sera renvoyé. Si

plus d'une sous-expression marquée est définie et mise en correspondance, seul le modèle capturé par la première sous-expression sera renvoyé. Si plus d'une sous-expression marquée est définie et que toutes ne correspondent pas, une chaîne vide sera renvoyée. Si aucune sous-expression n'est spécifiée, le modèle capturé par l'expression de recherche entière sera retourné.

Note: Lors de la saisie d'expressions régulières dans des scripts Picard, vous devez échapper une barre oblique inverse « \ », un signe dollar « \$ », une virgule « , » et les parenthèses gauche et droite « (» et «) » pour forcer Picard à ne pas les interpréter dans le cadre de la commande de script. Cela se fait en insérant une barre oblique inverse avant le caractère à échapper. Par exemple, l'expression régulière `^\s*([0-9,\.\.]*$)` devrait être saisie sous la forme `^\\s*\\([0-9\\,\\.\\.\\.]*\\)\\$`.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$rsearch(test, \disc 1\), \\(disc \(\d+\)) ) ==> "1"
$rsearch(test, \disc 1\), \\(disc \d+\\) ) ==> "(disc 1)"
$rsearch(test,x) ==> ""
$rsearch(test,t) ==> "t"
$rsearch(test,s) ==> "s"
$rsearch(test, \e\).*s) ==> "e"
$rsearch(test, \e\).*\s\\) ==> "e"
$rsearch(test, \e\).*x) ==> ""
$rsearch(test, \e\).*\(\x\\)) ==> ""
```

9.2.17 \$strip

Utilisation: **\$strip(texte)**

Catégorie: texte

La description:

Remplace tous les espaces dans `texte` par un seul espace et supprime les espaces de début et de fin. Les caractères d'espacement comprennent plusieurs espaces consécutifs et divers autres caractères Unicode. Les caractères tels que les retours à la ligne “\n”, les tabulations “\t” et les retours “\r” sont traités comme des espaces.

Exemple:

Les instructions suivantes renverront chacune « This text has been stripped. »:

```
$strip(This text has been stripped.)
$strip(This text has been stripped. )
$strip( This text has been stripped. )
$strip( This text has been stripped.)
```

(suite sur la page suivante)

(suite de la page précédente)

```
$strip( This text has been stripped. )
$strip(This text has been stripped.)
$strip(This text\rhas\nbeen\tstripped.)
```

9.2.18 \$substr

Utilisation: **\$substr(texte,début,fin)**

Catégorie: texte

Mis en œuvre: Picard 2.3

La description:

Renvoie la sous-chaîne de texte commençant par le caractère à l'index début, jusqu'à (mais non compris) le caractère à l'index fin. Les index sont basés sur zéro. Les nombres négatifs seront comptés à partir de la fin de la chaîne. Si les index de début ou de fin sont laissés vides, ils seront respectivement par défaut au début et à la fin de la chaîne. Si l'index début est évalué à un nombre négatif (par exemple: texte est « abc » et début est -10), il sera par défaut au début de la chaîne. De même, si l'index fin est un nombre supérieur au nombre de caractères de la chaîne, il sera par défaut à la fin de la chaîne. Les valeurs d'index invalides (par exemple: début supérieur à fin) renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$substr(abcdefg)      ==> "abcdefg"
$substr(abcdefg,3)    ==> "defg"
$substr(abcdefg,,3)   ==> "abc"
$substr(abcdefg,0,3)   ==> "abc"
$substr(abcdefg,-3)   ==> "efg"
$substr(abcdefg,-6,3)  ==> "bc"
$substr(abcdefg,-10,3) ==> "abc"
$substr(abcdefg,3,1)   ==> ""
```

9.2.19 \$swapprefix

Utilisation: **\$swapprefix(texte[,préfixes])**

Catégorie: texte

Implémenté: Picard 1.3 (*précédemment en tant que plugin depuis Picard 0.13*)

La description:

Déplace les préfixes spécifiés du début à la fin de texte. N'importe quel nombre de préfixes peut être spécifié, séparés par des virgules. Si aucun préfixe n'est spécifié,

« A » et « The » sont utilisés par défaut. Notez que la correspondance est sensible à la casse.

Exemple:

Si l “albumartiste est « Le Butcherettes », les instructions suivantes renverront les valeurs indiquées:

<code>\$swapprefix(%albumartist%)</code>	<code>==> "Le Butcherettes"</code>
<code>\$swapprefix(%albumartist%,le)</code>	<code>==> "Le Butcherettes"</code>
<code>\$swapprefix(%albumartist%,L)</code>	<code>==> "Le Butcherettes"</code>
<code>\$swapprefix(%albumartist%,A,An,The,Le)</code>	<code>==> "Butcherettes, Le"</code>

9.2.20 \$title

Utilisation: **\$title(texte)**

Catégorie: texte

Mis en œuvre: Picard 2.1

La description:

Renvoie le « texte » avec le premier caractère de chaque mot en majuscule. Notez que les autres caractères des mots ne seront pas modifiés, ce qui permet de conserver tous les acronymes en majuscules tels que « BBC ». Pour n’avoir que le premier caractère de chaque mot en majuscule, vous pourriez d’abord changer le texte en minuscule.

Exemples:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$set(foo,tHe houR is upOn uS)</code>	
<code>\$title(%foo%)</code>	<code>==> "THe HouR Is UpOn US"</code>
<code>\$title(\$lower(%foo%))</code>	<code>==> "The Hour Is Upon Us"</code>
<code>\$set(bar,THIS TEXT IS ALL CAPITALS)</code>	
<code>\$title(%bar%)</code>	<code>==> "THIS TEXT IS ALL CAPITALS"</code>
<code>\$title(\$lower(%bar%))</code>	<code>==> "This Text Is All Capitals"</code>
<code>\$set(baz,AC/DC recorded live at the BBC studio in London)</code>	
<code>\$title(%baz%)</code>	<code>==> "AC/DC Recorded Live At The BBC Studio In London"</code>
<code>\$title(\$lower(%baz%))</code>	<code>==> "Ac/Dc Recorded Live At The Bbc Studio In London"</code>

9.2.21 \$trim

Utilisation: **\$trim(texte[,caractère])**

Catégorie: texte

La description:

Supprime tous les espaces de début et de fin du texte. Le deuxième paramètre optionnel caractère spécifie le caractère à découper. Si plusieurs caractères sont fournis dans caractère, chaque caractère sera appliqué séparément à la fonction.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$trim( Trimmed )      ==> "Trimmed"  
$trim(__Trimmed__,_) ==> "Trimmed"  
$trim(x__Trimmed__y,_x) ==> "Trimmed__y"
```

9.2.22 \$truncate

Utilisation: **\$truncate(texte,longueur)**

Catégorie: texte

Mis en œuvre: Picard 0.12

La description:

Tronquez texte en longueur. Si longueur est inférieur à 0, alors la valeur utilisée est le nombre de caractères dans texte plus longueur (par exemple: \$truncate(abcd, -1) est le même comme \$truncate(abcd,3)). Si longueur est manquant ou un nombre négatif supérieur au nombre de caractères dans texte, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$truncate(Once upon a time,)      ==> ""  
$truncate(Once upon a time,0)      ==> ""  
$truncate(Once upon a time,3)      ==> "Onc"  
$truncate(Once upon a time,-3)     ==> "Once upon a t"  
$truncate(Once upon a time,-30)    ==> ""
```

9.2.23 \$upper

Utilisation: **\$upper(texte)**

Catégorie: texte

La description:

Renvoie texte en majuscules.

Exemple:

L'instruction suivante renverra la valeur indiquée:

```
$upper(This text is UPPER case) ==> "THIS TEXT IS UPPER CASE"
```

9.3 Fonctions à valeurs multiples

Ces fonctions permettent de gérer des balises ou des variables à valeurs multiples. Les fonctions de script à valeurs multiples sont:

9.3.1 \$getmulti

Utilisation: **\$getmulti(nom,index[,séparateur])**

Catégorie: multi-valeur

Mis en œuvre: Picard 2.3

La description:

Obtient l'élément à index à partir de la variable à valeurs multiples nom. Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour forcer la valeur en une valeur multiple appropriée variable.

L'index est basé sur zéro. Si index est inférieur à 0, alors la valeur utilisée est le nombre d'éléments dans nom plus index (par exemple: \$getmulti(%abcd%, -1) est identique à \$getmulti(%abcd%, 3) si %abcd% est une variable à valeurs multiples avec quatre éléments). Si index est manquant, pas un entier, un nombre supérieur ou égal au nombre d'éléments dans nom, ou un nombre négatif supérieur au nombre d'éléments dans nom, alors le fonction retournera une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,A; B; C)
$setmulti(bar,A; B; C)
$set(baz,1)
$getmulti(%foo%,%baz%)      ==> ""
$getmulti(%foo%,0)          ==> "A; B; C"
$getmulti(%foo%,-1)         ==> "A; B; C"
$getmulti(%foo%,-%baz%)    ==> "A; B; C"
$getmulti(%bar%,%baz%)     ==> "B"
$getmulti(%bar%,0)          ==> "A"
$getmulti(%bar%,-1)         ==> "C"
$getmulti(%bar%,-%baz%)    ==> "C"

$getmulti(A:1; B:2; C:3,1)  ==> "B:2"
$getmulti(A:1; B:2; C:3,1,:) ==> "1; B"
$getmulti(A:1; B:2; C:3,10) ==> ""
$getmulti(A:1; B:2; C:3,-10) ==> ""
$getmulti(A:1; B:2; C:3,1.5) ==> ""
$getmulti(A:1; B:2; C:3,a)  ==> ""
```

9.3.2 \$join

Utilisation: **\$join(nom, texte[, séparateur])**

Catégorie: multi-valeur

Mis en œuvre: Picard 2.3

La description:

Joint tous les éléments de la variable à valeurs multiples nom, en plaçant texte entre chaque élément et renvoie le résultat sous forme de chaîne. Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » si l'il n'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,First:A; Second:B)
$join(%foo%, >> )           ==> "First:A; Second:B"
$join(%foo%, >> ,:)         ==> "First >> A; Second >> B"

$setmulti(bar,First:A; Second:B)
$join(%bar%, >> )           ==> "First:A >> Second:B"
$join(%bar%, >> ,:)         ==> "First >> A; Second >> B"

$join(First:A; Second:B, )
$join(First:A; Second:B, >> ) ==> "First:ASecond:B"
$join(First:A; Second:B, >> ,:) ==> "First:A >> Second:B"
$join(First:A; Second:B, >> ,:) ==> "First >> A; Second >> B"
```

9.3.3 \$lenmulti

Utilisation: **\$lenmulti(nom[, séparateur])**

Catégorie: multi-valeur

La description:

Renvoie le nombre d'éléments dans la variable à valeurs multiples nom. Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable. Si nom est manquant, \$lenmulti renverra « 0 ». Si séparateur est spécifié mais laissé vide (par exemple \$setmulti(A; B; C,)), la fonction retournera « 1 ».

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$set(foo,)</code>	
<code>\$lenmulti(%foo%)</code>	==> "0"
<code>\$set(foo,A; B; C)</code>	
<code>\$lenmulti(%foo%)</code>	==> "1"
<code>\$setmulti(foo,A; B; C)</code>	
<code>\$lenmulti(%foo%)</code>	==> "3"
<code>\$lenmulti(A; B; C)</code>	==> "3"
<code>\$lenmulti(A:A; B:B; C:C, :)</code>	==> "4"
<code>\$lenmulti(,,)</code>	==> "0"
<code>\$lenmulti(,,:)</code>	==> "0"
<code>\$lenmulti(A; B; C,)</code>	==> "1"

9.3.4 \$map

Utilisation: **\$map(nom,code[, séparateur])**

Catégorie: multi-valeur

Mis en œuvre: Picard 2.3

La description:

Itère sur chaque élément trouvé dans la variable à valeurs multiples nom et met à jour la valeur de l'élément à la valeur renvoyée par code, renvoyant la variable à valeurs multiples mise à jour. Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable.

Pour chaque boucle, la valeur de l'élément est d'abord stockée dans la variable `_loop_value` et le décompte est stocké dans la variable `_loop_count`. Cela permet d'accéder à l'élément ou à la valeur de comptage dans le script de code.

Les éléments vides sont automatiquement supprimés de la sortie.

Note: Vous ne pouvez pas enregistrer le code dans une variable puis passer la variable à la fonction en tant que `%code%` car il sera évalué lorsqu'il sera affecté à la variable plutôt que pendant la boucle.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,First:A; Second:B)
$map(%foo%,
    $upper(%_loop_count%=%_loop_value%))    ==> "1=FIRST:A; SECOND:B"
$map(%foo%,
    $upper(%_loop_count%=%_loop_value%),:) ==> "1=FIRST:2=A; SECOND:3=B"

$setmulti(bar,First:A; Second:B)
$map(%bar%,
    $upper(%_loop_count%=%_loop_value%))    ==> "1=FIRST:A; 2=SECOND:B"
$map(%bar%,
    $upper(%_loop_count%=%_loop_value%),:) ==> "1=FIRST:2=A; SECOND:3=B"

$setmulti(baz,A; B; x; C)
$map(%baz%,$if($eq(%_loop_value%,x),,%_loop_count%=%_loop_value%))
                ==> "1=A; 2=B; 4=C"
$map(First:A; Second:B,
    $upper(%_loop_count%=%_loop_value%))    ==> "1=FIRST:A; 2=SECOND:B"
```

9.3.5 \$performer

Utilisation: **\$performer(motif[, séparateur])**

Catégorie: multi-valeur

Mis en œuvre: Picard 0.10

La description:

Renvoie les interprètes dont le type de performance correspond à motif séparés par séparateur (ou une virgule suivie d'un espace « , » si non passé). Si motif est vide, tous les interprètes seront renvoyés. Notez que le motif à mettre en correspondance est sensible à la casse.

Exemple:

Avec les balises performer comme `performer:guitar = « Ann »`, `performer:rhythm-guitar = « Bob »` et `performer:drums = « Cindy »`, les

instructions suivantes renverront les valeurs indiquées:

```
$set(foo,guitar)
$performer(%foo%)      ==> "Ann, Bob"

$performer(guitar)    ==> "Ann, Bob"
$performer(Guitar)    ==> ""
$performer(rhythm-guitar) ==> "Bob"
$performer()           ==> "Ann, Bob, Cindy"
$performer(, / )       ==> "Ann / Bob / Cindy"
```

9.3.6 \$replacemulti

Utilisation: **\$replacemulti(nom,recherche,replace[, séparateur])**

Catégorie : multi-valeurs

Mis en œuvre: Picard 2.6.1

Description:

Remplace les occurrences de recherche par replace dans la variable à valeurs multiples nom et renvoie la chaîne de variables à valeurs multiples résultante avec les éléments séparés par séparateur (ou le séparateur par défaut d'un point-virgule suivi d'un espace « ; » si n'est pas passé).

Les éléments vides sont automatiquement supprimés de la sortie.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$setmulti(foo,Electronic; Idm; Techno)
$replacemulti(%foo%,Idm,IDM)      ==> "Electronic; IDM; Techno"

$setmulti(foo,Electronic; Jungle; Bardcore)
$replacemulti(%foo%,Bardcore,Hardcore) ==> "Electronic; Jungle; Hardcore
↪"

$setmulti(foo,One; Two; Three)
$replacemulti(%foo%,Four,Five)      ==> "One; Two; Three"

$setmulti(foo,Four; Five; Six)
$replacemulti(%foo%,Five,)          ==> "Four; Six"
```

9.3.7 \$reversemulti

Utilisation: **\$reversemulti(nom[, séparateur])**

Catégorie: multi-valeur

Mis en œuvre: Picard 2.3.1

La description:

Renvoie une copie de la variable à valeurs multiples nom avec les éléments dans l'ordre inverse. Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable.

Cette fonction peut être utilisée en conjonction avec la fonction [\\$sortmulti](#) pour trier par ordre décroissant.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,A; B; C; D; E)
$reversemulti(%foo%)          ==> "A; B; C; D; E"

$setmulti(bar,A; B; C; D; E)
$reversemulti(%bar%)          ==> "E; D; C; B; A"

$setmulti(baz,A:A; B:B; C:C,:)
$reversemulti(%baz%)          ==> "C; B; C; A; B; A"

$reversemulti(A; B; C; D; E)   ==> "E; D; C; B; A"
$reversemulti(A:A; B:B; C:C,:) ==> "C:B; C:A; B:A"
```

9.3.8 \$slice

Utilisation: **\$slice(nom,début,fin[, séparateur])**

Catégorie: multi-valeur

Mis en œuvre: Picard 2.3

La description:

Renvoie une variable à valeurs multiples contenant les éléments de l'index début jusqu'à mais sans inclure l'index fin de la variable à valeurs multiples nom. Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable.

Les index sont basés sur zéro. Les nombres négatifs seront comptés à partir du nombre d'éléments de la liste. Si les index début ou fin sont laissés vides, ils seront par défaut

à 0 et au nombre d'éléments dans la liste respectivement.

Une utilisation typique pourrait être de créer une variable à valeurs multiples avec tous les artistes dans %artists% sauf le premier, qui peut être utilisé pour créer un « feat. » liste. Cela ressemblerait à quelque chose comme \$setmulti(\$support_artists, \$slice(%artists%,1,)).

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

\$set(foo,A; B; C; D; E)	==> " "
\$slice(%foo%,1)	
\$setmulti(foo,A; B; C; D; E)	==> "B; C; D; E"
\$slice(%foo%,1)	
\$slice(A; B; C; D; E,1)	==> "B; C; D; E"
\$slice(A; B; C; D; E,1,3)	==> "B; C"
\$slice(A; B; C; D; E,,3)	==> "A; B; C"
\$slice(A; B; C; D; E,1,3)	==> "B; C"
\$slice(A; B; C; D; E,1,-1)	==> "B; C; D"
\$slice(A; B; C; D; E,-4,4)	==> "B; C; D"
\$slice(A:A; B:B; C:C; D:D; E:E,,1,:)	==> "A"
\$slice(A:A; B:B; C:C; D:D; E:E,-2,,:)	==> "D; E:E"
\$slice(A:A; B:B; C:C; D:D; E:E,2,4,:)	==> "B; C:C; D"

9.3.9 \$sortmulti

Utilisation: **\$sortmulti(nom[, séparateur])**

Catégorie: multi-valeur

Mis en œuvre: Picard 2.3.1

La description:

Renvoie une copie de la variable à valeurs multiples nom avec les éléments triés par ordre croissant. Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » si l'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable. Si nom est manquant, \$sortmulti renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

\$set(foo,B; C; E; D; A)	==> "B; C; E; D; A"
\$sortmulti(%foo%)	
\$setmulti(foo,B; C; E; D; A)	==> "A; B; C; D; E"
\$sortmulti(%foo%)	

(suite sur la page suivante)

(suite de la page précédente)

```
$sortmulti(B; D; E; A; C)      ==> "A; B; C; D; E"  
$sortmulti(B:AB; D:C; E:D; A:A; C:X,:) ==> "A; C:AB; D:B:C; E:D; A:X"  
$sortmulti(,)                  ==> ""  
$sortmulti(,:)                  ==> ""
```

9.3.10 \$unique

Utilisation : **\$unique(*nom*[,*sensible_à_la_casse*[,*séparateur*]])**

Catégorie : multi-valeurs

Implémenté : Picard 2.6.1

Description:

Retourne une copie triée de la variable à valeurs multiples *nom* avec les éléments en double supprimés. Par défaut, la comparaison ne tient pas compte de la casse des éléments ; cependant, cela peut être modifié en donnant à *sensible_à_la_casse* une valeur non vide. Une valeur littérale représentant une multi-valeur peut être substituée à *nom*, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour contraindre la valeur dans une variable multi-valeur appropriée. Si *nom* est manquant, \$unique retournera une chaîne vide.

Note: Lors d'une comparaison insensible à la casse (par défaut), le dernier élément correspondant sera utilisé dans le résultat. Par exemple, si la variable à valeurs multiples contient “abc”, “Abc”, “ABc” et “ABC” dans cet ordre, l’élément “ABC” sera inclus dans le résultat.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$setmulti(foo,a; A; B; b; cd; Cd; cD; CD; a; A; b)  
$set(bar,a; A; B; b; cd; Cd; cD; CD; a; A; b)  
  
$unique(%foo%)    ==> "A; CD; b"  
$unique(%bar%)    ==> "a; A; B; b; cd; Cd; cD; CD; a; A; b"  
$unique(%foo%,1)  ==> "A; B; CD; Cd; a; b; cD; cd"  
  
$unique(a; A; B; b; cd; Cd; cD; CD; a; A; b) ==> "A; CD; b"
```

9.4 Fonctions mathématiques

Ces fonctions sont utilisées pour effectuer des opérations arithmétiques sur des balises ou des variables. Les fonctions de script mathématiques sont:

9.4.1 \$add

Utilisation: **\$add(x,y,*args)**

Catégorie: mathématique

La description:

Ajoute y à x. Peut être utilisé avec un nombre arbitraire d'arguments (ex: $\$add(x, y, z) = (x + y) + z$). Renvoie une chaîne vide si un argument est manquant ou pas un entier.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$add(20,15)      ==> "35"
$add(20,-15)     ==> "5"
$add(20,14,1)    ==> "35"
$add(20,10,3,2)  ==> "35"
$add(20,10,3,)   ==> ""
$add(20,10,3,a)  ==> ""
$add(20,10,3.5)  ==> ""
```

9.4.2 \$div

Utilisation: **\$div(x,y,*args)**

Catégorie: mathématique

La description:

Divise x par y et renvoie la valeur entière (arrondie vers le bas). Peut être utilisé avec un nombre arbitraire d'arguments (par exemple: $\$div(x, y, z) = (x / y) / z$). Si un argument est vide ou n'est pas un entier, la fonction renverra une chaîne vide. Si le deuxième argument ou tout argument suivant est zéro, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$div(10,3)      ==> "3"
$div(10,-3)     ==> "-4"
$div(-10,3)     ==> "-4"
$div(10,3,2)    ==> "1"
$div(10,-3,-2)  ==> "2"
$div(10,2,1.5)  ==> ""
$div(10,2,0)    ==> ""
$div(10,2,x)   ==> ""
$div(10,2,)     ==> ""
```

9.4.3 \$mod

Utilisation: **\$mod(x,y,*args)**

Catégorie: mathématique

La description:

Renvoie le reste de x divisé par y. Peut être utilisé avec un nombre arbitraire d'arguments (ex: $\$mod(x,y,z) = (x \% y) \% z$). Si un argument est vide ou n'est pas un entier, la fonction renverra une chaîne vide. Si le deuxième argument ou tout argument suivant est zéro, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$mod(0,3)      ==> "0"
$mod(10,3)     ==> "1"
$mod(10,-3)    ==> "-2"
$mod(-13,10)   ==> "7"
$mod(13,-10)   ==> "-7"
$mod(10,3,1)   ==> "0"
$mod(50,17,9)  ==> "7"
$mod(51,3,0)   ==> ""
$mod(51,a)     ==> ""
$mod(a,10)     ==> ""
$mod(,10)       ==> ""
$mod(10,)       ==> ""
$mod(10,3.5)   ==> ""
```

9.4.4 \$mul

Utilisation: **\$mul(x,y,*args)**

Catégorie: mathématique

La description:

Multiplie x par y. Peut être utilisé avec un nombre arbitraire d'arguments (c'est-à-dire: $\$mul(x,y,z) = (x * y) * z$). Si un argument est vide ou n'est pas un entier, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$mul(1,2)      ==> "2"
$mul(1,2,3)    ==> "6"
$mul(1,2,0)    ==> "0"
$mul(1,-2,3)   ==> "-6"
$mul(-1,2,-3)  ==> "6"
$mul(1,2,)      ==> ""
$mul(1,2,x)    ==> ""
$mul(1,2,5)    ==> ""
```

9.4.5 \$sub

Utilisation: **\$sub(x,y,*args)**

Catégorie: mathématique

La description:

Soustrait y de x. Peut être utilisé avec un nombre arbitraire d'arguments (ex: $\$sub(x,y,z) = (x - y) - z$). Renvoie une chaîne vide si un argument est manquant ou pas un entier.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$sub(20,15)    ==> "5"
$sub(20,-15)   ==> "35"
$sub(20,14,1)  ==> "5"
$sub(20,10,3,2) ==> "5"
$sub(20,10,3,)  ==> ""
$sub(20,10,3,a) ==> ""
$sub(20,10,3,5) ==> ""
```

9.5 Fonctions conditionnelles

Ces fonctions sont utilisées pour tester diverses conditions et prendre les mesures appropriées en fonction des résultats du test. Les fonctions de script conditionnelles sont:

9.5.1 \$and

Utilisation: **\$and(x,y,*args)**

Catégorie: conditionnel

La description:

Renvoie vrai si les deux x et y ne sont pas vides. Peut être utilisé avec un nombre arbitraire d'arguments. Le résultat est vrai si **TOUS** des arguments ne sont pas vides.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(test,x)
$and(%test%,)      ==>  ""  (False)
$and(%test%,1)    ==>  "1"  (True)
$and(%test%,A)    ==>  "1"  (True)
$and(%test%,$gt(4,5)) ==>  ""  (False)
$and(%test%,$lt(4,5)) ==>  "1"  (True)
$and(%test%,,)     ==>  ""  (False)
$and(%test%,,0)    ==>  ""  (False)
$and(%test%,, )    ==>  ""  (False)
$and(%test%, , )   ==>  "1"  (True)
```

9.5.2 \$endswith

Utilisation: **\$endswith(texte, suffixe)**

Catégorie: conditionnel

Mis en œuvre: Picard 1.4

La description:

Renvoie vrai si texte se termine par suffixe. Notez que la comparaison est sensible à la casse.

Exemple:

Les instructions ci-dessous renvoient les valeurs indiquées:

```
$endswith(The time is now,is now) ==> "1" (True)
$endswith(The time is now,is NOW) ==> "" (False)
$endswith(The time is now,)      ==> "1" (True)
$endswith(,.)                  ==> "1" (True)
$endswith(,now)                ==> "" (False)
```

9.5.3 \$eq

Utilisation: **\$eq(x,y)**

Catégorie: conditionnel

Mis en œuvre: Picard

La description:

Renvoie vrai si x est égal à y. Notez que les comparaisons sont sensibles à la casse.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$eq(,a)    ==> "" (False)
$eq(a,)    ==> "" (False)
$eq(a,A)   ==> "" (False)
$eq(a,a)   ==> "1" (True)
```

9.5.4 \$eq_all

Utilisation: **\$eq_all(x,a1,a2,*args)**

Catégorie: conditionnel

La description:

Renvoie vrai si x est égal à a1 et a2, etc. Peut être utilisé avec un nombre arbitraire d'arguments. Notez que les comparaisons sont sensibles à la casse.

Fonctionnellement équivalent à `$and($eq(x,a1),$eq(x,a2) ...)`.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$eq_all(A,A,B,C)    ==> "" (False)
$eq_all(A,a,A,A)    ==> "" (False)
$eq_all(A,A,A,A)    ==> "1" (True)
$eq_all(,,,)        ==> "1" (True)
$eq_all(,a,)         ==> "" (False)
```

9.5.5 \$eq_any

Utilisation: **\$eq_any(x,a1,a2,*args)**

Catégorie: conditionnel

La description:

Renvoie vrai si x est égal à a1 ou a2, etc. Peut être utilisé avec un nombre arbitraire d'arguments. Notez que les comparaisons sont sensibles à la casse.

Fonctionnellement équivalent à `$or($eq(x,a1),$eq(x,a2) ...)`.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$eq_any(A,A,B,C) ==> "1" (True)
$eq_any(A,a,A,A) ==> "1" (True)
$eq_any(A,a,b,c) ==> "" (False)
$eq_any(,,,) ==> "1" (True)
$eq_any(,a,b,c) ==> "" (False)
```

9.5.6 \$gt

Utilisation: **\$gt(x,y)**

Catégorie: conditionnel

Mis en œuvre: Picard

La description:

Renvoie vrai si x est supérieur à y. Si un argument est manquant ou n'est pas un entier, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$gt(-1,0) ==> "" (False)
$gt(0,0) ==> "" (False)
$gt(1,0) ==> "1" (True)
$gt(,) ==> "" (False)
$gt(,0) ==> "" (False)
$gt(0,) ==> "" (False)
$gt(a,1) ==> "" (False)
$gt(1,a) ==> "" (False)
$gt(1,1.5) ==> "" (False)
```

9.5.7 \$gte

Utilisation: **\$gte(x,y)**

Catégorie: conditionnel

Mis en œuvre: Picard

La description:

Renvoie vrai si x est supérieur ou égal à y. Si un argument est manquant ou n'est pas un entier, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$gte(-1,0)    ==>  ""   (False)
$gte(0,0)    ==>  "1"  (True)
$gte(1,0)    ==>  "1"  (True)
$gte(, )     ==>  ""   (False)
$gte(,0)     ==>  ""   (False)
$gte(0,)     ==>  ""   (False)
$gte(a,1)    ==>  ""   (False)
$gte(1,a)    ==>  ""   (False)
$gte(1,1.5)   ==>  ""   (False)
```

9.5.8 \$if

Utilisation: **\$if(condition,then[,else])**

Catégorie: conditionnel

La description:

Si condition n'est pas vide, elle renvoie then, sinon elle retourne else. Si else n'est pas fourni, il sera supposé être une chaîne vide. En plus (ou au lieu de) renvoyer des valeurs, then et else peuvent être utilisés pour exécuter conditionnellement d'autres fonctions.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$set(foo,This is foo)
$set(bar, )
$if(%foo%,%foo%,No foo)           ==>  "This is foo"
$if(%bar%,%bar%,No bar)          ==>  "No bar"
$if(%bar%,This is bar,No bar)    ==>  "No bar"
$if(%bar%,This is bar,)          ==>  ""
$if(%bar%,This is bar)           ==>  ""
```

(suite sur la page suivante)

(suite de la page précédente)

<code>\$if(,True,False)</code>	<code>==> "False"</code>
<code>\$if(,True,False)</code>	<code>==> "True"</code>
<code>\$if(,\$set(value,True),\$set(value,False))</code>	<code>==> Sets "value" to "False"</code>
<code>\$set(value,\$if(%bar%,True,False))</code>	<code>==> Sets "value" to "False"</code>

9.5.9 \$if2

Utilisation: `$if2(a1,a2,a3,...)`

Catégorie: conditionnel

La description:

Renvoie le premier argument non vide. Peut être utilisé avec un nombre arbitraire d'arguments.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$set(foo,)</code>	
<code>\$set(bar,Something)</code>	
<code>\$if2(%foo%,%bar%,Three)</code>	<code>==> "Something"</code>
<code>\$if2(,%bar%,Three)</code>	<code>==> "Something"</code>
<code>\$if2(,%foo%,%bar%,Three)</code>	<code>==> "Something"</code>
<code>\$if2(%foo%, ,%bar%,Three)</code>	<code>==> " "</code>
<code>\$if2(%foo%.,%bar%,Three)</code>	<code>==> ". "</code>
<code>\$if2(%foo%, ,Three)</code>	<code>==> "Three"</code>
<code>\$if2(%foo%,,,)</code>	<code>==> ""</code>

9.5.10 \$in

Utilisation: `$in(x,y)`

Catégorie: conditionnel

Mis en œuvre: Picard

La description:

Renvoie vrai, si x contient y. Notez que les comparaisons sont sensibles à la casse.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$set(foo,ABCDEFG)</code>	
<code>\$set(bar,CDE)</code>	
<code>\$in(%foo%,%bar%)</code>	<code>==> "1" (True)</code>

(suite sur la page suivante)

(suite de la page précédente)

```
$in(ABCDE,CDE)    ==> "1"  (True)
$in(ABCDE,CE)     ==> ""   (False)
$in(ABCDE,cde)    ==> ""   (False)
$in(ABCDE,)        ==> "1"  (True)
$in(,)             ==> "1"  (True)
```

9.5.11 \$inmulti

Utilisation: **\$inmulti(%x%,y)**

Catégorie: conditionnel

Mis en œuvre: Picard 1.0

La description:

Renvoie true si la variable à valeurs multiples x contient exactement y comme l'une de ses valeurs. Notez que les comparaisons sont sensibles à la casse.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$setmulti(foo,One; Two; Three)
$set(bar,Two)
$inmulti(%foo%,%bar%) ==> "1"  (True)
$inmulti(%foo%,Two)    ==> "1"  (True)
$inmulti(%foo%,two)    ==> ""   (False)
$inmulti(%foo%,Once)   ==> ""   (False)
$inmulti(%foo%,w)      ==> ""   (False)
$inmulti(%foo%,)        ==> ""   (False)
```

9.5.12 \$is_audio

Utilisation: **\$is_audio()**

Catégorie: conditionnel

Mis en œuvre: Picard 2.2

La description:

Renvoie true, si la piste en cours de traitement n'est pas affichée comme étant une vidéo.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$is_audio() ==> "1" (True, if the track is not a video)
$is_audio() ==> "" (False, if the track is a video)
```

9.5.13 \$is_complete

Utilisation: **\$is_complete()**

Catégorie: conditionnel

La description:

Renvoie true si chaque piste de l'album correspond à un seul fichier.

Note: Cette fonction ne fonctionne que dans les scripts d'attribution de noms de fichiers.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$is_complete() ==> "1" (True, if all tracks have been matched)
$is_complete() ==> "" (False, if not all tracks have been matched)
```

9.5.14 \$is_video

Utilisation: **\$is_video()**

Catégorie: conditionnel

Mis en œuvre: Picard 2.2

La description:

Renvoie true, si la piste en cours de traitement est affichée comme étant une vidéo.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$is_video() ==> "1" (True, if the track is a video)
$is_video() ==> "" (False, if the track is not a video)
```

9.5.15 \$lt

Utilisation: **\$lt(x,y)**

Catégorie: conditionnel

La description:

Renvoie vrai si x est inférieur à y. Si un argument est manquant ou n'est pas un entier, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$lte(-1,0)  ==> "1"  (True)
$lte(0,0)   ==> ""   (False)
$lte(1,0)   ==> ""   (False)
$lte(,0)    ==> ""   (False)
$lte(,0)    ==> ""   (False)
$lte(0,)    ==> ""   (False)
$lte(a,1)   ==> ""   (False)
$lte(1,a)   ==> ""   (False)
$lte(1,1.5) ==> ""   (False)
```

9.5.16 \$lte

Utilisation: **\$lte(x,y)**

Catégorie: conditionnel

La description:

Renvoie vrai si x est inférieur ou égal à y. Si un argument est manquant ou n'est pas un entier, la fonction renverra une chaîne vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$lte(-1,0)  ==> "1"  (True)
$lte(0,0)   ==> "1"  (True)
$lte(1,0)   ==> ""   (False)
$lte(,0)    ==> ""   (False)
$lte(,0)    ==> ""   (False)
$lte(0,)    ==> ""   (False)
$lte(a,1)   ==> ""   (False)
$lte(1,a)   ==> ""   (False)
$lte(1,1.5) ==> ""   (False)
```

9.5.17 \$ne

Utilisation: **\$ne(x,y)**

Catégorie: conditionnel

La description:

Renvoie vrai si x n'est pas égal à y. Notez que les comparaisons sont sensibles à la casse.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$ne(,a)    ==> "1"  (True)
$ne(a,)    ==> "1"  (True)
$ne(a,A)   ==> "1"  (True)
$ne(a,a)   ==> ""   (False)
```

9.5.18 \$ne_all

Utilisation: **\$ne_all(x,a1,a2,*args)**

Catégorie: conditionnel

La description:

Renvoie vrai si x n'est pas égal à a1 et a2, etc. Peut être utilisé avec un nombre arbitraire d'arguments. Notez que les comparaisons sont sensibles à la casse.

Fonctionnellement équivalent à `$and($ne(x,a1),$ne(x,a2) ...)`.

Exemple:

The following statements will return the values indicated:

```
$ne_all(A,A,B,C)    ==> ""  (False)
$ne_all(A,a,A,A)    ==> ""  (False)
$ne_all(A,a,a,a)    ==> "1" (True)
$ne_all(,,,)         ==> ""  (False)
$ne_all(,a,a)        ==> "1" (True)
```

9.5.19 \$ne_any

Utilisation: **\$ne_any(x,a1,a2,*args)**

Catégorie: conditionnel

La description:

Renvoie vrai si x n'est pas égal à a1 ou a2, etc. Peut être utilisé avec un nombre arbitraire d'arguments. Notez que les comparaisons sont sensibles à la casse.

Fonctionnellement équivalent à `$or($ne(x,a1),$ne(x,a2) ...)`.

Exemple:

The following statements will return the values indicated:

<code>\$ne_any(A,A,B,C)</code>	<code>==> "1" (True)</code>
<code>\$ne_any(A,a,A,A)</code>	<code>==> "1" (True)</code>
<code>\$ne_any(A,A,A,A)</code>	<code>==> "" (False)</code>
<code>\$ne_any(,,,)</code>	<code>==> "" (False)</code>
<code>\$ne_any(,a,,)</code>	<code>==> "1" (True)</code>

9.5.20 \$not

Utilisation: **\$not(x)**

Catégorie: conditionnel

La description:

Renvoie vrai si x est vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$set(foo,)</code>	
<code>\$not(%foo%)</code>	<code>==> "1" (False)</code>
<code>\$not(x)</code>	<code>==> "" (True)</code>
<code>\$not()</code>	<code>==> "" (True)</code>
<code>\$not()</code>	<code>==> Error</code>

9.5.21 \$or

Utilisation: **\$or(x,y,*args)**

Catégorie: conditionnel

La description:

Renvoie vrai si x ou y n'est pas vide. Peut être utilisé avec un nombre arbitraire d'arguments. Le résultat est vrai si **TOUT** des arguments n'est pas vide.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$or(,)      ==> ""  (False)
$or(,1)     ==> "1" (True)
$or(,A)     ==> "1" (True)
$or(,$gt(4,5)) ==> ""  (False)
$or(,$lt(4,5)) ==> "1" (True)
$or(,,)     ==> ""  (False)
$or(,,0)    ==> "1" (True)
$or(,, )    ==> "1" (True)
```

9.5.22 \$startswith

Utilisation: **\$startswith(texte,préfixe)**

Catégorie: conditionnel

Mis en œuvre: Picard 1.4

La description:

Renvoie vrai si texte commence par préfixe. Notez que la comparaison est sensible à la casse.

Exemple:

Les instructions ci-dessous renvoient les valeurs indiquées:

```
$startswith(The time is now.,The time) ==> "1" (True)
$startswith(The time is now.,The TIME) ==> ""  (False)
$startswith(The time is now.,)       ==> "1" (True)
$startswith(,The)                  ==> ""  (False)
$startswith(,)                     ==> "1" (True)
```

9.6 Fonctions d'information

Ces fonctions fournissent des informations supplémentaires sur le système ou les données. Les fonctions de script d'informations sont:

9.6.1 \$datetime

Utilisation: **\$datetime([format])**

Catégorie: information

Mis en œuvre: Picard 2.3

La description:

Renvoie la date et l'heure actuelles dans le format spécifié, qui est basé sur le code de format standard Python strftime <<https://strftime.org>>`. Si aucun format n'est spécifié, la date et l'heure seront renvoyées sous la forme "2020-02-15 14:26:32". Notez que tous les caractères spéciaux tels que "%", "\$", "(", ")" et "\\" devront être échappés comme indiqué dans les exemples ci-dessous.

Avertissement: Les codes de formatage spécifiques à la plate-forme doivent être évités pour garantir la portabilité des scripts sur les différentes plates-formes. Ces codes incluent: supprimer zéro-padding (par exemple: %-d et %-m sous Linux ou macOS, et leurs équivalents %#d et %#m sur Windows); spécificateurs de longueur d'élément (par exemple: %3Y); et accrocher "%" à la fin de la chaîne de format.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

<code>\$datetime()</code>	<code>==> "2020-02-15 14:26:32"</code>
<code>\$datetime(\%Y-\%m-\%d \%H:\%M:\%S)</code>	<code>==> "2020-02-15 14:26:32"</code>
<code>\$datetime(\%Y-\%m-\%d)</code>	<code>==> "2020-02-15"</code>
<code>\$datetime(\%H:\%M:\%S)</code>	<code>==> "14:26:32"</code>
<code>\$datetime(\%B \%d, \%Y)</code>	<code>==> "February 15, 2020"</code>

9.6.2 \$matchedtracks

Utilisation: **\$matchedtracks()**

Catégorie: information

Mis en œuvre: Picard 0.12

La description:

Renvoie le nombre de pistes correspondantes dans une version.

Note: Cette fonction ne fonctionne que dans les scripts d'attribution de noms de fichiers.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$matchedtracks() ==> "3" (if three of the tracks were matched)
```

9.7 Fonctions de boucle

Ces fonctions offrent la possibilité de répéter des actions en fonction du contenu d'une variable à valeurs multiples ou du résultat d'un test conditionnel. Les fonctions de script de boucle sont:

9.7.1 \$foreach

Utilisation: **\$foreach(nom,code,séparateur= »; »)**

Catégorie: boucle

Mis en œuvre: Picard 2.3

La description:

Itère sur chaque élément trouvé dans la variable à valeurs multiples nom, exécutant code à chaque itération. Avant chaque itération, la valeur de l'élément est d'abord stockée dans la variable `_loop_value` et le décompte est stocké dans la variable `_loop_count`. Cela permet d'accéder à l'élément ou à la valeur de comptage dans le script de code.

Une valeur littérale représentant une valeur multiple peut être remplacée par nom, en utilisant le séparateur (ou un point-virgule suivi d'un espace « ; » s'il n'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable.

Exemple:

Les instructions suivantes effectueront le traitement indiqué:

```
$noop( Mark all listed tags for deletion from the files. )
$foreach(genre; comment; year,$delete(%_loop_value%))

$noop( Create an 'artist_count' tag with a count of all artists
      listed for the track. )
$foreach(%artists%, $set(artist_count,%_loop_count%))

$noop( Create a separate tag for each artist listed for the
```

(suite sur la page suivante)

(suite de la page précédente)

```
track as 'artist_1', 'artist_2', etc. )
$foreach(%artists%, $set(artist_%_loop_count%, %_loop_value%))
```

9.7.2 \$while

Utilisation: **\$while(condition,code)**

Catégorie: boucle

Mis en œuvre: Picard 2.3

La description:

Exécute le code à plusieurs reprises jusqu'à ce que la condition n'évalue plus à True. Pour chaque boucle, le décompte est stocké dans la variable _loop_count. Cela permet d'accéder à la valeur de comptage dans le script code.

Note: La fonction limite le nombre maximum d'itérations à 1000 pour éviter de créer accidentellement une boucle infinie.

Exemple:

L'instruction suivante définira return sur « Echo ... echo ... echo ... »:

```
$set(return,Echo...) $while($lt(%_loop_count%,2), $set(return,%return% echo...))
```

9.8 Fonctions diverses

Les diverses fonctions de script sont:

9.8.1 \$noop

Utilisation: **\$noop(...)**

Catégorie: divers

La description:

Ne fait rien et renvoie toujours une chaîne vide. Ceci est utile pour les commentaires ou pour désactiver un bloc de code.

Exemple:

Les instructions suivantes renverront les valeurs indiquées:

```
$noop( A comment. )      ==>  ""
$noop($set(foo,Testing...)) ==>  "" (and "foo" is not set)
```

CHAPTER 10

Utiliser Picard

Il y a quatre étapes pour utiliser Picard pour traiter vos fichiers audio:

10.1 Récupération des informations sur l'album

Cette étape identifie l'album sur MusicBrainz qui fournira les informations utilisées pour baliser les fichiers, et récupère les métadonnées de la base de données MusicBrainz. Il existe différentes méthodes disponibles, en fonction des informations actuellement disponibles sur votre système (par exemple: les métadonnées existant dans les fichiers, ou la disponibilité du CD source).

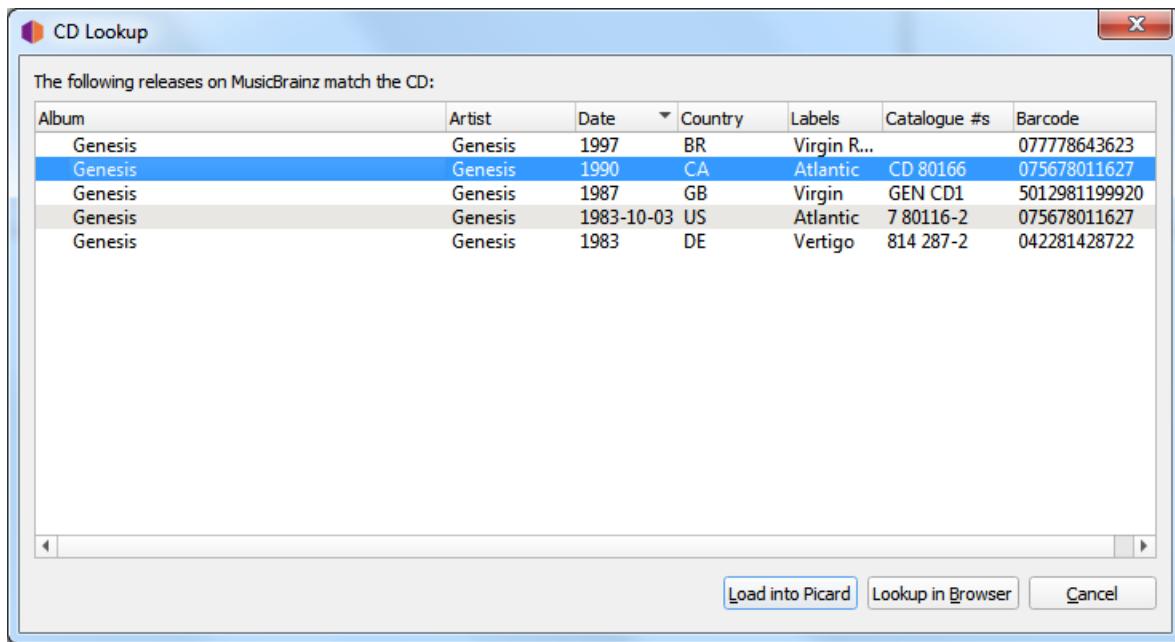
Il existe essentiellement quatre méthodes principales utilisées pour récupérer les informations d'album à partir de la base de données MusicBrainz.

10.1.1 CD de recherche

Il s'agit de la méthode préférée pour identifier automatiquement l'album à récupérer et doit être utilisée lorsque le CD est disponible. En règle générale, cela serait utilisé juste après l'extraction des fichiers audio du CD. Une fois lancée, la table des matières (toc) est lue à partir du CD et une demande est envoyée à MusicBrainz pour renvoyer une liste des versions qui correspondent à la toc. S'il y a des correspondances, elles seront répertoriées pour que vous puissiez sélectionner celle à utiliser. S'il n'y a pas de correspondance ou si aucune des correspondances n'est correcte, vous pouvez rechercher manuellement dans la base de données l'album correspondant et vous avez la possibilité de joindre le toc de votre CD à la version sélectionnée pour une recherche ultérieure.

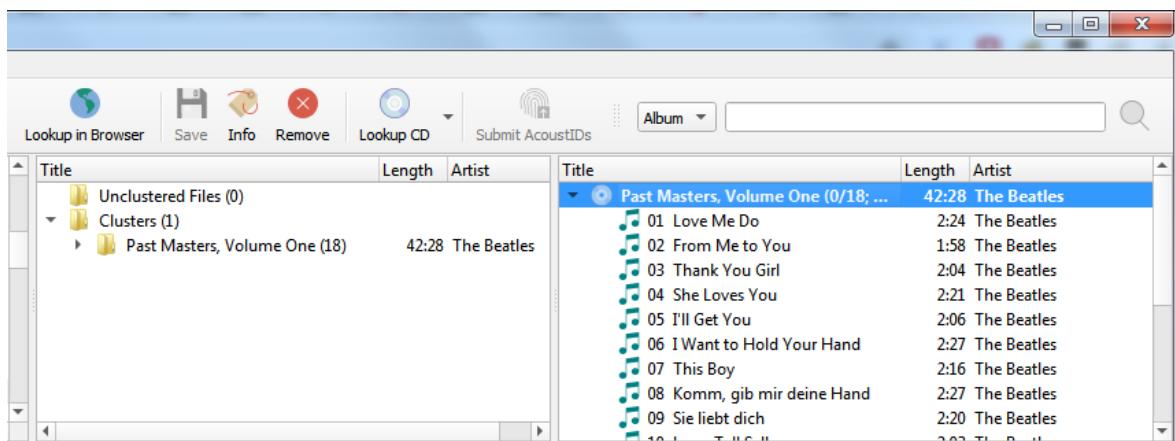
Les étapes à suivre pour rechercher un CD sont:

- Assurez-vous que le CD est inséré dans le lecteur et sélectionnez « Outils → Rechercher un CD... → (lecteur à utiliser) ». Le toc du CD sera calculé et envoyé à MusicBrainz, et une liste des versions correspondantes sera affichée.

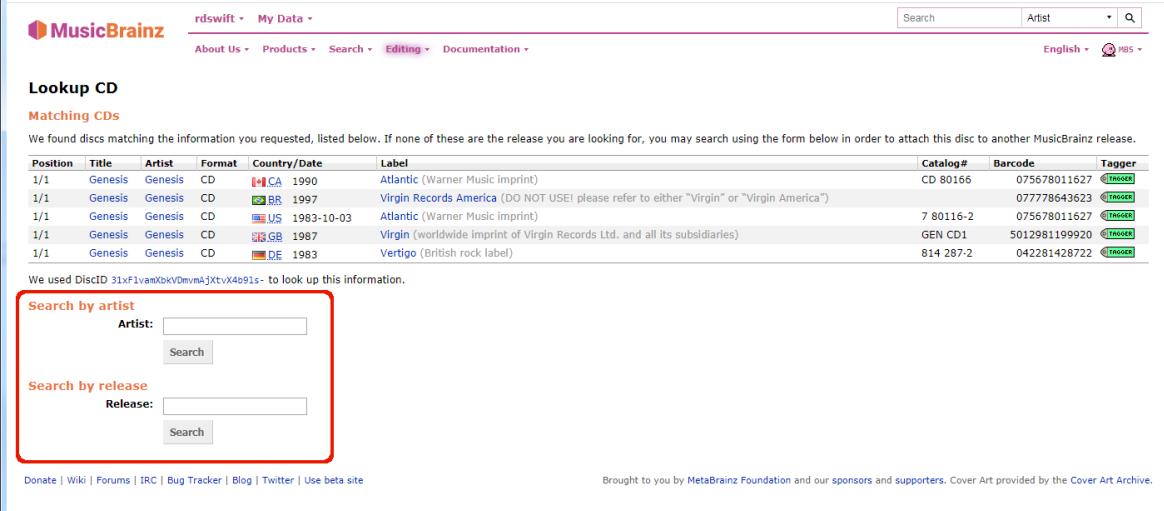


- Sélectionnez la version correcte dans la liste. Cela chargera les informations pour la publication dans Picard.

Un symbole de musique devant un numéro de piste dans le volet de droite indique qu'aucun fichier n'a été attribué à la piste.



3. S'il n'y a pas de correspondance ou si aucune des correspondances n'est correcte, utilisez l'option « *Rechercher dans le navigateur* » pour localiser la version correcte. Saisissez le titre de la sortie ou l'artiste et lancez la recherche. Vous recevrez une liste des versions qui correspondent à votre critère de recherche et qui ont le même nombre de pistes que votre CD toc.



We found discs matching the information you requested, listed below. If none of these are the release you are looking for, you may search using the form below in order to attach this disc to another MusicBrainz release.

Position	Title	Artist	Format	Country/Date	Label	Catalog#	Barcode	Tagger
1/1	Genesis	Genesis	CD	CA 1990	Atlantic (Warner Music imprint)	CD 80166	075678011627	
1/1	Genesis	Genesis	CD	BR 1997	Virgin Records America (DO NOT USE! please refer to either "Virgin" or "Virgin America")		077778643623	
1/1	Genesis	Genesis	CD	US 1983-10-03	Atlantic (Warner Music imprint)	7 80116-2	075678011527	
1/1	Genesis	Genesis	CD	GB 1987	Virgin (worldwide imprint of Virgin Records Ltd. and all its subsidiaries)	GEN CD1	5012981199920	
1/1	Genesis	Genesis	CD	DE 1983	Vertigo (British rock label)	814 287-2	042281428722	

We used DiscID 31x1vam0k\UmmdjxtvX4b91s- to look up this information.

Search by artist
Artist:

Search by release
Release:

[Donate](#) | [Wiki](#) | [Forums](#) | [IRC](#) | [Bug Tracker](#) | [Blog](#) | [Twitter](#) | [Use beta site](#)

Brought to you by [MetaBrainz Foundation](#) and our sponsors and [supporters](#). Cover Art provided by the [Cover Art Archive](#).

4. Utilisez la flèche verte pour charger les informations pour une publication dans Picard. De plus, vous pouvez sélectionner la version et attacher la toc.



<input type="radio"/> CD 1 (show tracklist) Genesis	Genesis	JP 1985-06-01	Vertigo (British rock label)	32PD-17	4988011303999	
<input type="radio"/> CD 1 (show tracklist) Genesis	Genesis	NL 1983-10-03	Mercury Records (or just "Mercury". A UMG imprint, do not use it for ©/© credits)	832 178-1	042281428715	
<input type="radio"/> 12" Vinyl 1 (show tracklist) Genesis	Genesis	CA 1990	Atlantic (Warner Music imprint)	CD 80166	07567801127	
<input checked="" type="radio"/> CD 1 (show tracklist) Genesis (made in Japan)	Genesis	XE - US -	Atlantic (Warner Music imprint)	7 80116-2, 814287-2	[none]	
<input type="radio"/> CD 1 (show tracklist)						

5. Si aucune des versions affichées n'est correcte, vous avez la possibilité d'ajouter une nouvelle version (avec certaines informations automatiquement incluses).

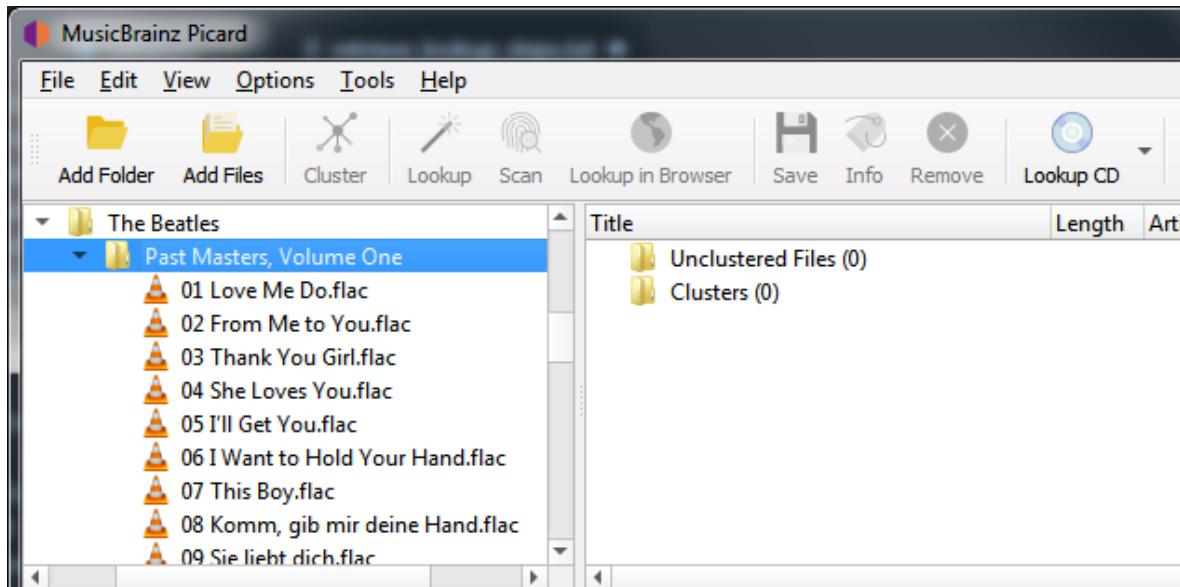
The screenshot shows the MusicBrainz Picard web interface. At the top, there is a navigation bar with links for 'CD 1 (show tracklist)', 'Attach CD TOC', and 'Add a new release'. Below this, a section titled 'Add a new release' contains the text: 'If you don't see the release you are looking for, you can still add a new one, using this CD TOC:' followed by a button labeled 'Add a new release'. At the bottom of the page, there is a footer with links for 'Donate | Wiki | Forums | IRC | Bug Tracker | Blog | Twitter | Use beta site'.

10.1.2 Rechercher des fichiers

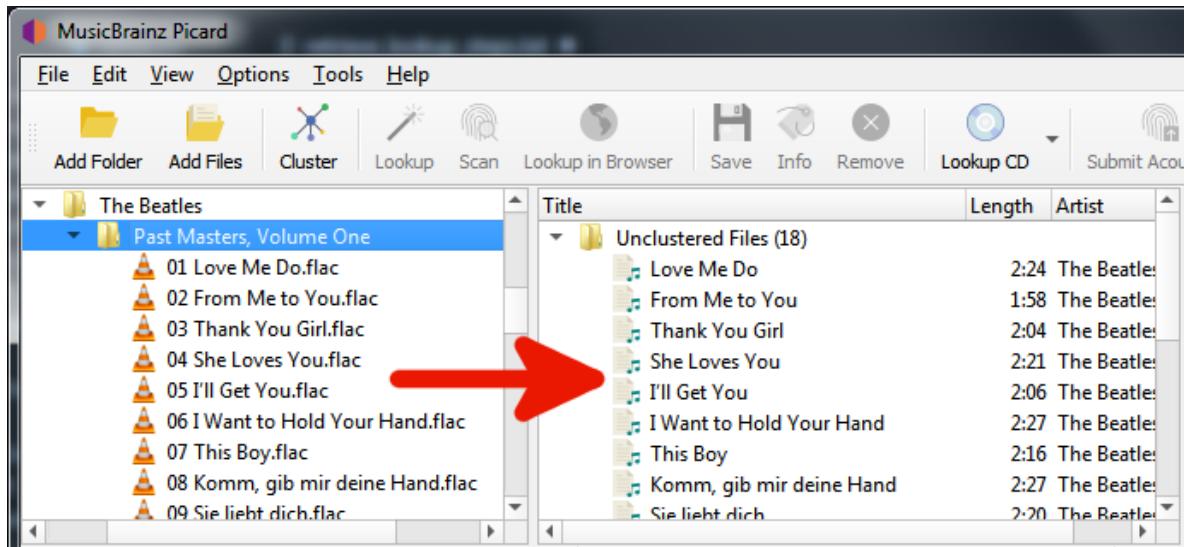
Si vous ne disposez pas du CD et que vos fichiers sont regroupés par album, il s'agit de la méthode préférée pour identifier automatiquement l'album à récupérer. Cela se fait en regroupant les fichiers dans des groupes d'albums dans Picard, puis en effectuant la recherche. Picard essaiera de faire correspondre l'ensemble des fichiers en cluster à la même version.

Les étapes à suivre pour rechercher des fichiers sont:

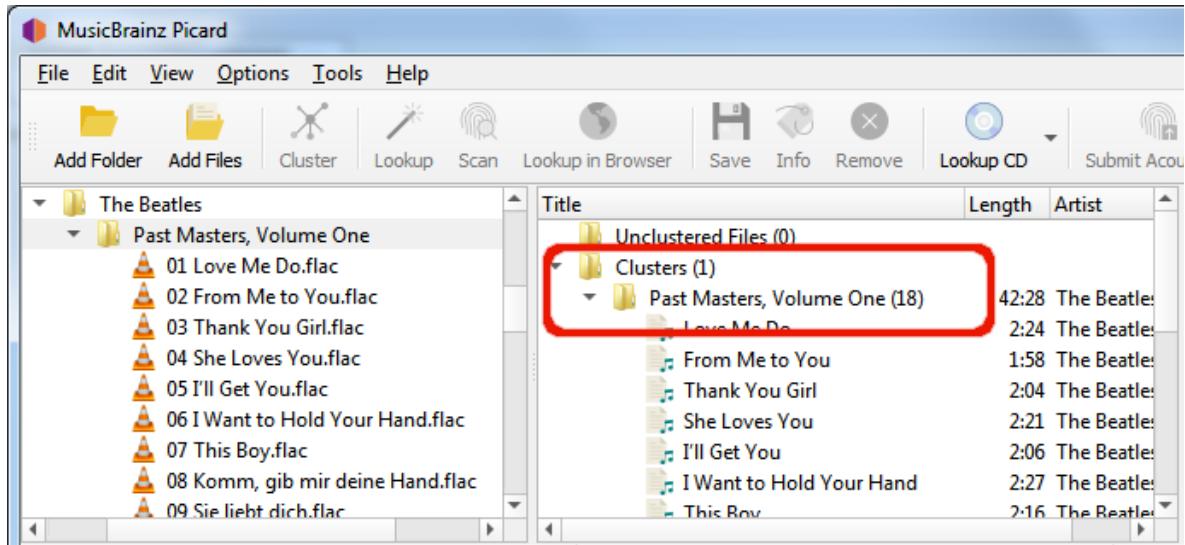
1. Ajoutez vos fichiers en utilisant « *Fichiers → Ajouter des fichiers...* » ou « *Fichiers → Ajouter un dossier...* ». Pour faciliter l'utilisation, il est recommandé d'utiliser le navigateur de fichiers interne pour gérer les interactions du système de fichiers. Ceci est activé à partir de « *Affichage → Navigateur de fichiers* ».



- Faites glisser le répertoire ou les fichiers sélectionnés vers le dossier « Fichiers non groupés » et attendez que Picard traite les fichiers - les noms passeront du gris au noir.



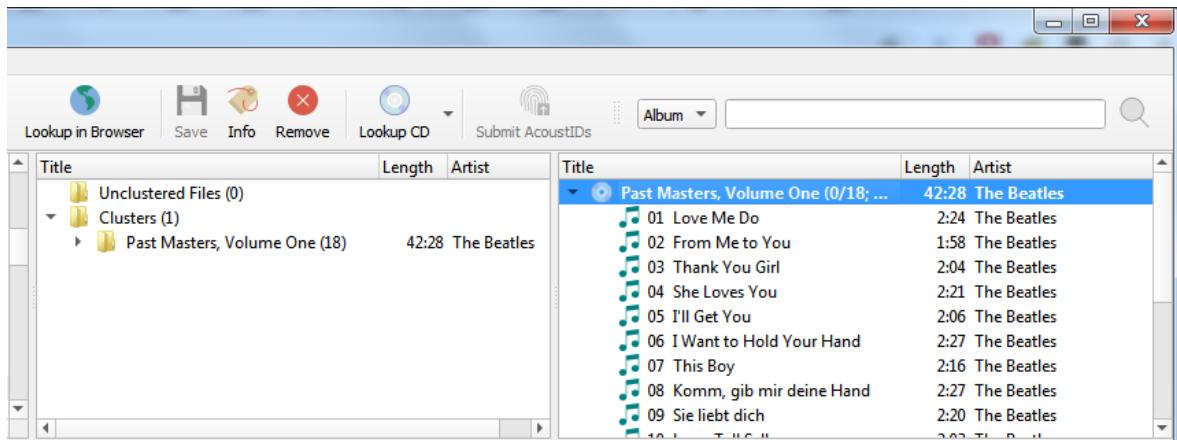
- Utilisez « Outils → Cluster » pour regrouper les fichiers en groupes d'albums.



- Sélectionnez un album en cluster et utilisez « Outils → Rechercher » pour rechercher le cluster. En fonction de vos métadonnées précédentes, l'album

apparaîtra dans le volet de droite.

Un symbole de musique devant un numéro de piste dans le volet de droite indique qu'aucun fichier n'a été attribué à la piste.



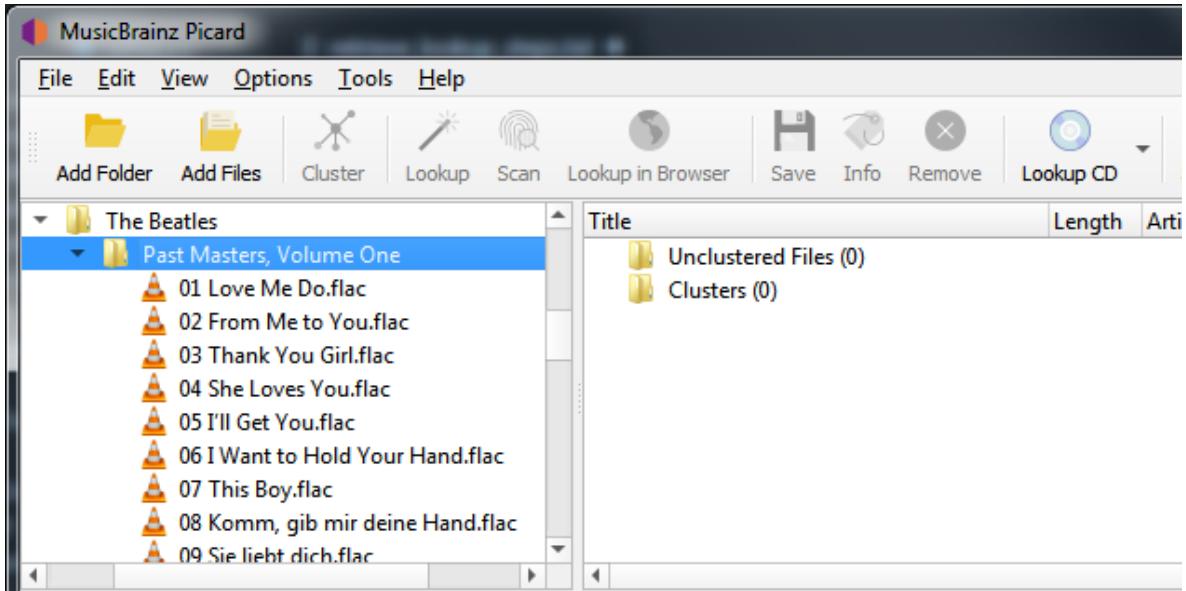
Si aucun album n'a été récupéré, ou si l'album récupéré était incorrect, vous devrez peut-être essayer une autre méthode telle que l'analyse des fichiers ou une recherche manuelle.

10.1.3 Analyser les fichiers

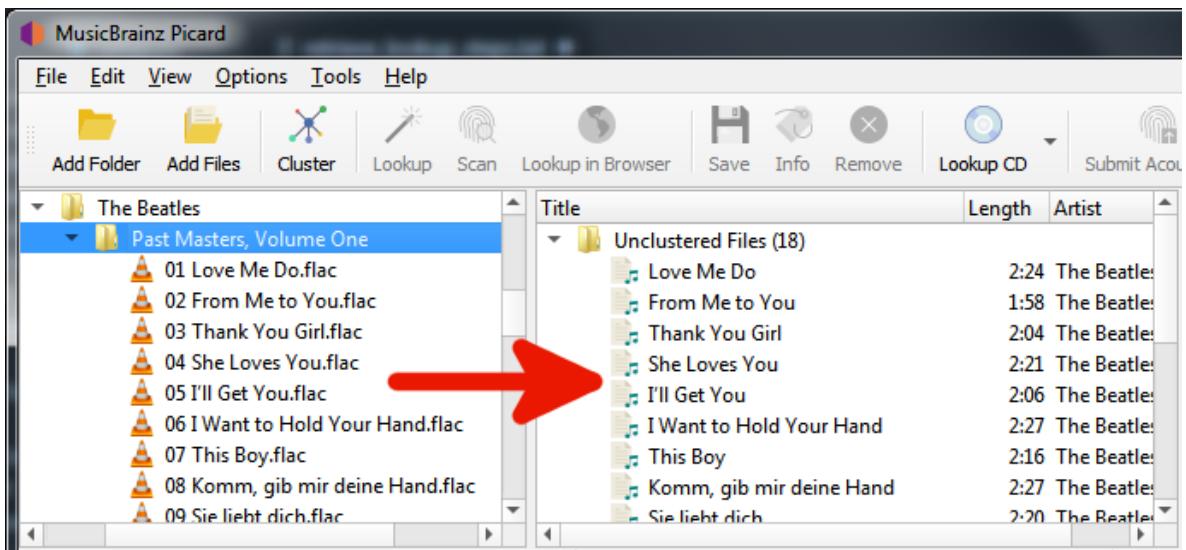
Si vos fichiers ne sont pas regroupés en albums et que vous ne disposez pas du CD, c'est la seule méthode restante pour identifier automatiquement l'album à récupérer. Cela se fait en scannant les fichiers pour obtenir leurs empreintes digitales AcoustID, puis en effectuant la recherche des fichiers individuels par empreinte digitale. Les albums correspondant aux fichiers s'afficheront dans le volet de droite en fonction d'une « meilleure correspondance » en utilisant les paramètres de versions préférées dans les options de métadonnées.

Les étapes à suivre pour analyser et rechercher des fichiers sont les suivantes:

1. Ajoutez vos fichiers en utilisant « *Fichiers → Ajouter des fichiers...* » ou « *Fichiers → Ajouter un dossier...* ». Pour faciliter l'utilisation, il est recommandé d'utiliser le navigateur de fichiers interne pour gérer les interactions du système de fichiers. Ceci est activé à partir de « *Affichage → Navigateur de fichiers* ».



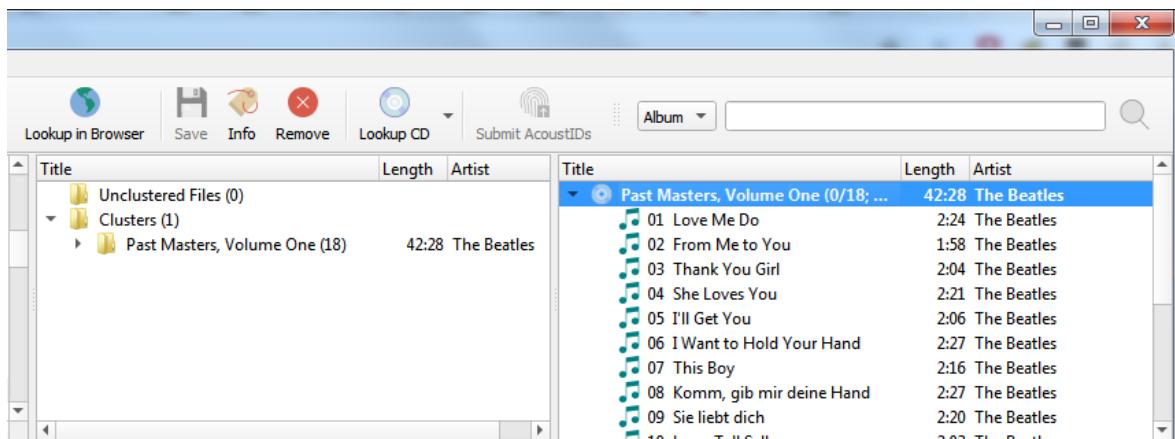
- Faites glisser le répertoire ou les fichiers sélectionnés vers le dossier « Fichiers non groupés » et attendez que Picard traite les fichiers - les noms passeront du gris au noir.



- Sélectionnez les fichiers souhaités et utilisez « Outils → Analyser » pour scanner les fichiers afin de déterminer leurs empreintes AcoustID et rechercher à l'aide de

ces informations. Le ou les albums correspondant aux fichiers s'afficheront dans le volet de droite en fonction d'une «meilleure correspondance» en utilisant les paramètres de versions préférées dans les options de métadonnées.

Un symbole de musique devant un numéro de piste dans le volet de droite indique qu'aucun fichier n'a été attribué à la piste.



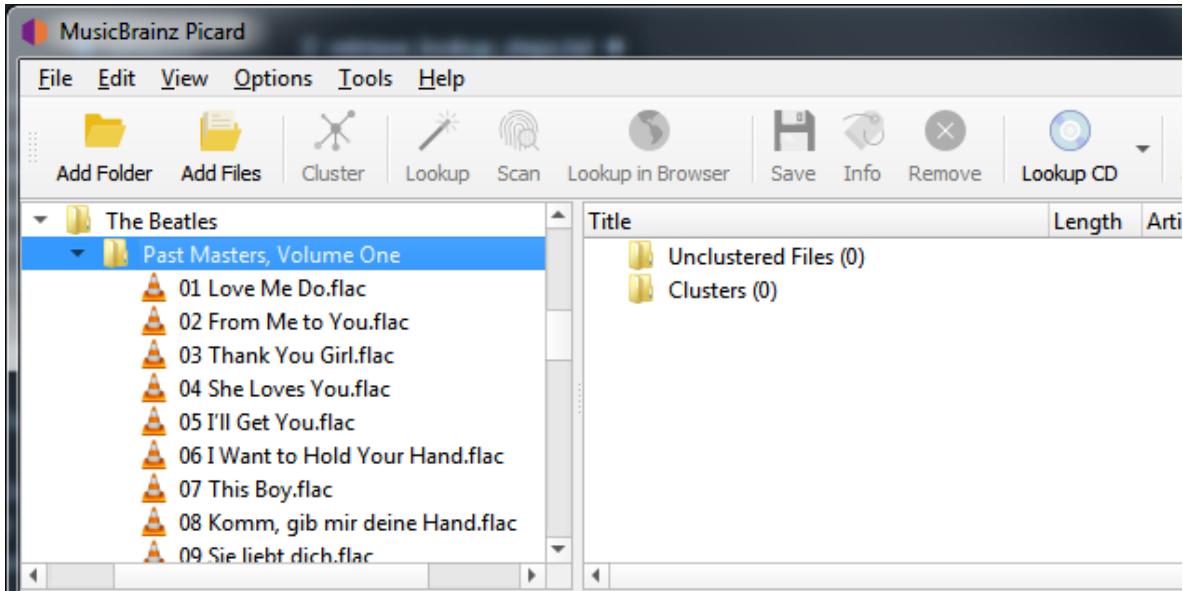
Si aucun album n'a été récupéré ou si l'album récupéré était incorrect, vous devrez peut-être essayer une autre méthode, telle que le regroupement des fichiers ou une recherche dans le navigateur.

10.1.4 Recherche dans le navigateur

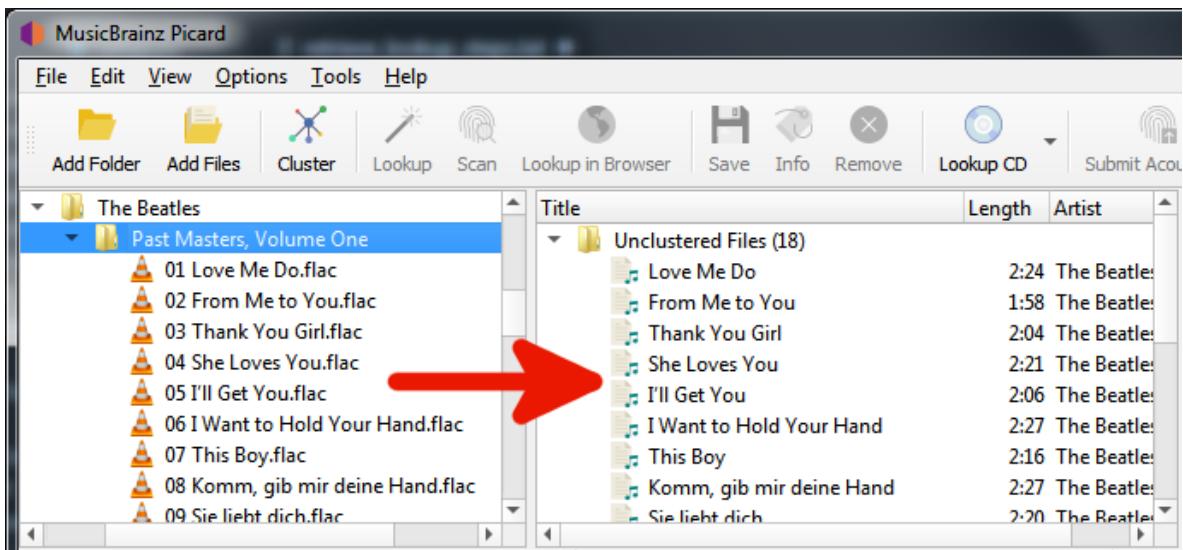
Si aucune des méthodes automatisées n'est disponible ou ne produit pas les résultats souhaités, vous avez la possibilité de récupérer les informations de l'album en demandant à Picard de lancer une recherche sur le site Web MusicBrainz à l'aide de votre navigateur Web. Il existe deux méthodes pour lancer cette recherche. La première méthode recherche sur la base des informations de balise des fichiers sélectionnés.

Les étapes à suivre pour rechercher manuellement un album sur MusicBrainz sont:

1. Ajoutez vos fichiers en utilisant « *Fichiers → Ajouter des fichiers...* » ou « *Fichiers → Ajouter un dossier...* ». Pour faciliter l'utilisation, il est recommandé d'utiliser le navigateur de fichiers interne pour gérer les interactions du système de fichiers. Ceci est activé à partir de « *Affichage → Navigateur de fichiers* ».

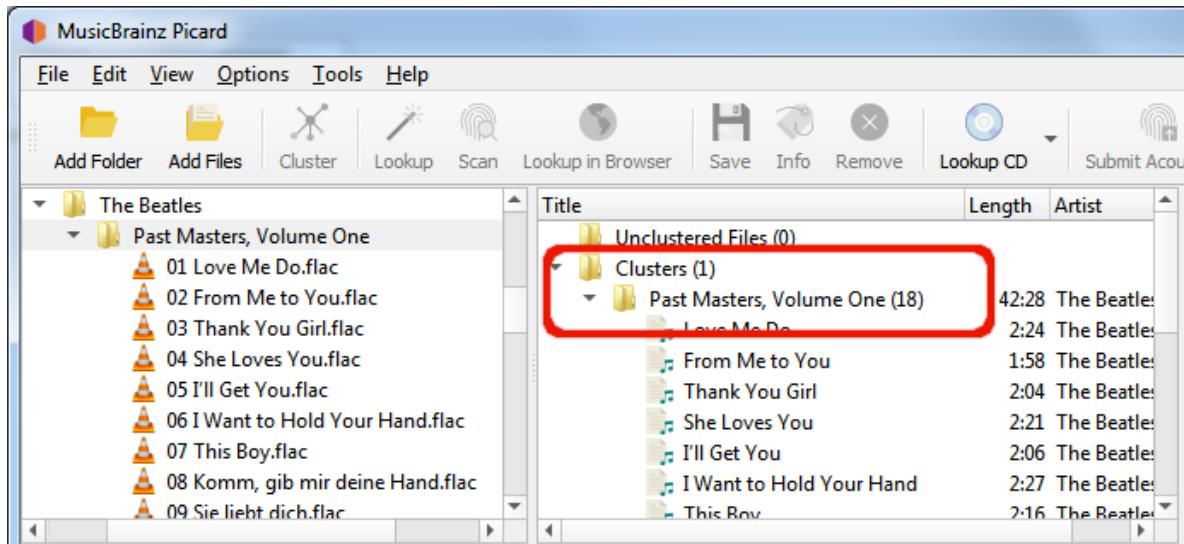


2. Faites glisser le répertoire ou les fichiers sélectionnés vers le dossier « Fichiers non groupés » et attendez que Picard traite les fichiers - les noms passeront du gris au noir.



3. Utilisez « Outils → Cluster » pour regrouper les fichiers en clusters d'albums si vous voulez rechercher un cluster.

MusicBrainz Picard, Version v2.6.3



4. Sélectionnez un fichier ou un album groupé et utilisez « Outils → Recherche dans le navigateur » pour lancer la recherche dans votre navigateur en utilisant les métadonnées actuellement disponibles.

The screenshot shows the MusicBrainz website at 'Tag Lookup Results'. The search term 'Past Masters, Volume One' is entered in the search bar. The results table has columns for Name, Artist, Format, Tracks, Country/Date, Label, Catalog#, Barcode, Language, Type, Status, and Tagger. There are 2,321 results found. The results show multiple entries for 'Past Masters, Volume One' by 'The Beatles' in various formats (CD, LP) and countries (UK, US, CA, JP). Some results have pink highlights, likely indicating specific matches or clusters.

Name	Artist	Format	Tracks	Country/Date	Label	Catalog#	Barcode	Language	Type	Status	Tagger
Past Masters, Volume One	The Beatles	CD	18	UK 1988	Parlophone	CD-BPM 1, CDP 7 90043 2	077779004324	eng / Latn	Album + Compilation	Official	TRIGGER
Past Masters, Volume One	The Beatles	CD	18	US 1988-03-08	Capitol Records, Parlophone	CDP 7 90043 2	077779004324	eng / Latn	Album + Compilation	Official	TRIGGER
Past Masters, Volume One	The Beatles	CD	18	CA 1988	Apple Records, Parlophone	C2 90043	077779004324	eng / Latn	Album + Compilation	Official	TRIGGER
Past Masters, Volume One	The Beatles	CD	18	JP -	EMI, Odeon	TOCP-8515, CP32-5601		eng / Latn	Album + Compilation	Official	TRIGGER
Past Masters, Volume One	The Beatles	CD	18	GB 1988-03-07	Parlophone	CD-BPM1, CDP 7 90043 2	077779004324	eng / Latn	Album + Compilation	Official	TRIGGER
Past Masters, Volume One	The Beatles	CD	18	JP 1998-03-11	EMI, Parlophone, Apple Records	TOCP-51125	4988006740082	eng / Latn	Album + Compilation	Official	TRIGGER
Past Masters, Volume One	The Beatles	CD	18	CA 1988	Parlophone	CDP 7 90043 2	077779004324	eng / Latn	Album + Compilation	Official	TRIGGER

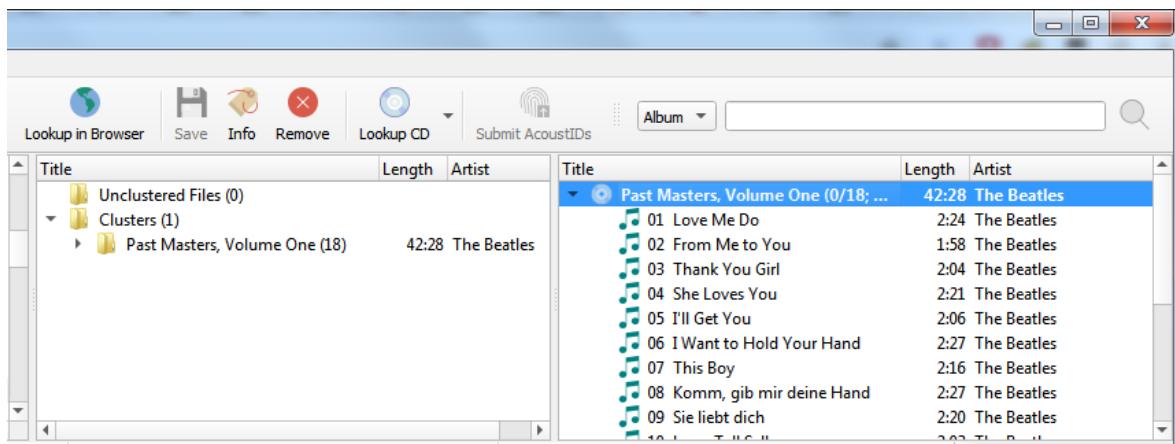
5. Si vous souhaitez réviser ou affiner vos critères de recherche, effectuez les modifications souhaitées en bas de la page Web et cliquez sur le bouton « Rechercher » pour relancer la recherche.

The screenshot shows a list of tracks for the song "All You Need Is Love" by The Beatles. The first track is "All You Need Is Love" (unknown) 2. Below it are "Let It Be" (7" Vinyl 2), "Free as a Bird" (unknown) 2, and "Nothing Is Real: The Making of "Strawberry Fields Forever"" (unknown) 14. To the right of the tracks is a detailed view of the first track, showing artist, release, and track information. A red box highlights the "Artist: The Beatles" field and the "Release: Past Masters, Volume One" field. Below these fields are track number, track, duration, and filename input boxes, followed by a "Search" button.

6. Utilisez la flèche verte pour charger les informations pour une publication dans Picard.

The screenshot shows a list of 2,321 results for "Past Masters, Volume One" by The Beatles. The results are listed in a table with columns for Name, Artist, Format, Tracks, Country/Date, Label, Catalog#, Barcode, Language, Type, Status, and Tagger. The third result in the list has a red box around its status column, which is labeled "Official".

7. Un symbole de musique devant un numéro de piste dans le volet de droite indique qu'aucun fichier n'a été attribué à la piste.

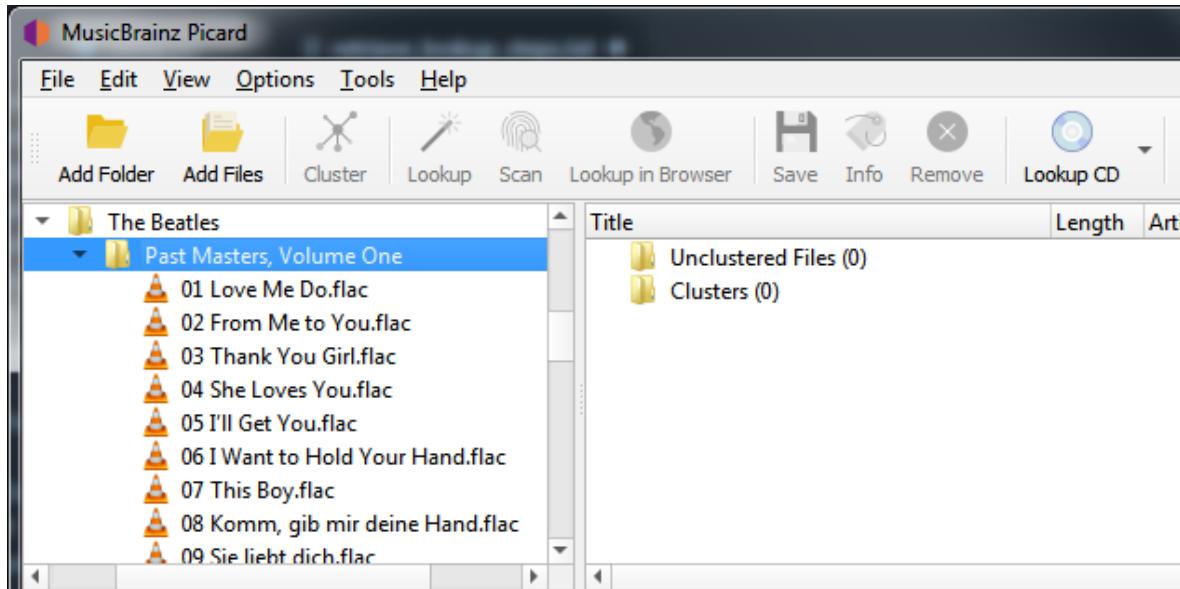


10.1.5 Recherche manuelle

La deuxième méthode de recherche par navigateur utilise des informations saisies manuellement comme critère de recherche.

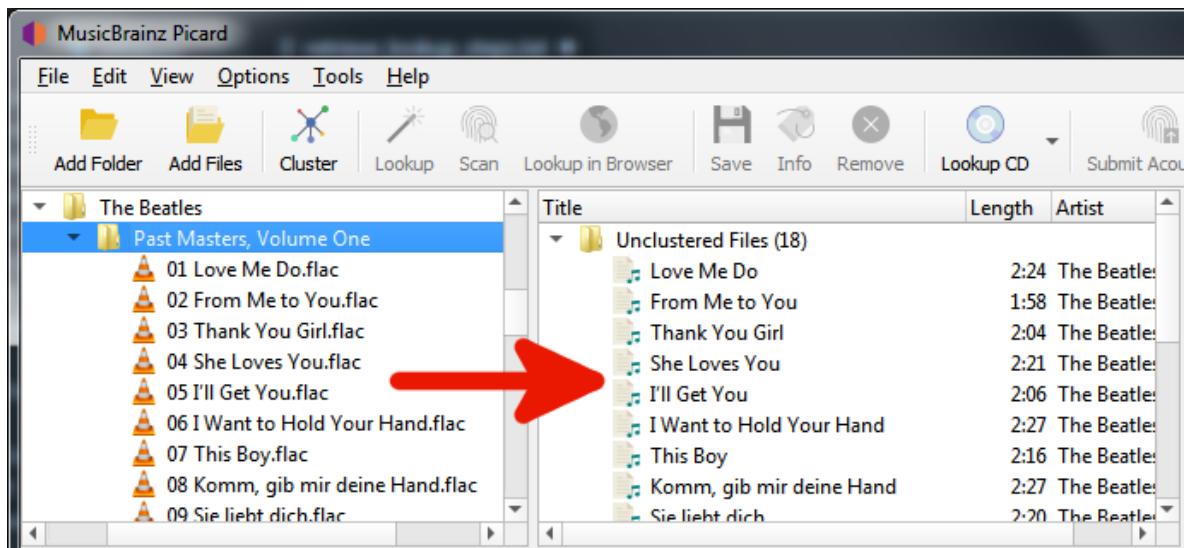
Les étapes à suivre pour rechercher manuellement un album sur MusicBrainz sont:

1. Ajoutez vos fichiers en utilisant « *Fichiers → Ajouter des fichiers...* » ou « *Fichiers → Ajouter un dossier...* ». Pour faciliter l'utilisation, il est recommandé d'utiliser le navigateur de fichiers interne pour gérer les interactions du système de fichiers. Ceci est activé à partir de « *Affichage → Navigateur de fichiers* ».



2. Faites glisser le répertoire ou les fichiers sélectionnés vers le dossier « *Fichiers non groupés* » et attendez que Picard traite les fichiers - les noms passeront du

gris au noir.



3. Entrez vos informations de recherche dans la zone de recherche et sélectionnez le type d'enregistrements à rechercher, puis cliquez sur le symbole de la loupe pour lancer la recherche. Cela ouvrira le site [Web MusicBrainz](#) dans votre navigateur.



4. Continuez à explorer en cliquant sur les liens appropriés jusqu'à ce que vous arriviez à la version que vous souhaitez récupérer.

MusicBrainz Picard, Version v2.6.3

The screenshot shows the MusicBrainz search results for the query "beatles". The results table has columns for Name, Sort Name, Type, Gender, Area, Begin, Begin Area, End, and End Area. The first result, "The Beatles", is highlighted with a red box.

Name	Sort Name	Type	Gender	Area	Begin	Begin Area	End	End Area
The Beatles	Beatles, The	Group		United Kingdom	1957-03	Liverpool	1970-04-10	
The Tape-beatles (US multimedia group)	Tape-beatles, The	Group		United States	1986-12	Iowa City		
Sex Beatles (Brazilian rock band)	Sex Beatles	Group						
Blues Beatles (blues rock, Brazil)	Blues Beatles	Group		Brazil				
Tokyo Beatles (Japanese)	Tokyo Beatles			Japan		Tokyo		
Beatles Chillout	Beatles Chillout							
Beatles Back2Back (The Beatles tribute band)	Beatles Back2Back	Group		Australia	2011			
Counterfeit Beatles (The Beatles tribute band)	Counterfeit Beatles	Group		United Kingdom				

- Utilisez la flèche verte pour charger les informations pour une publication dans Picard.

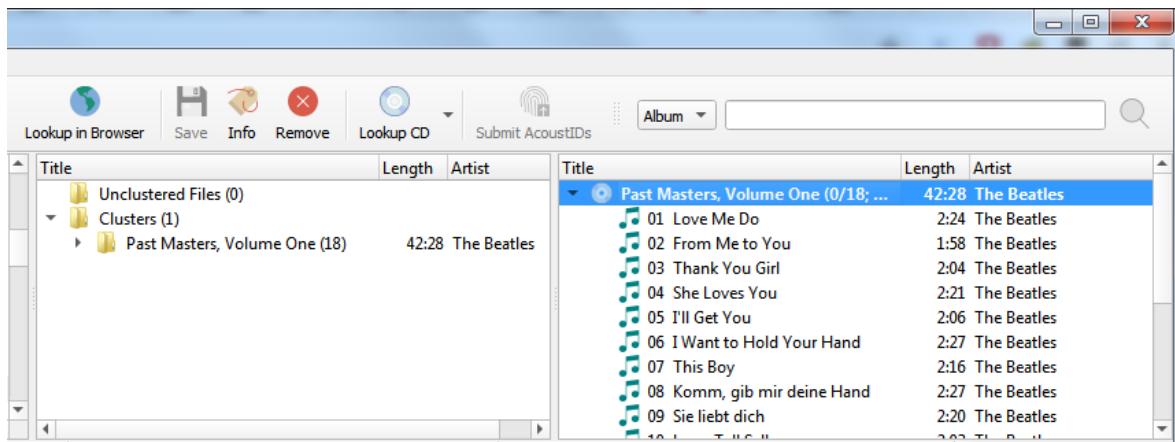
The screenshot shows the MusicBrainz release group page for "Past Masters, Volume One". The page includes a table of tracks and a sidebar with release group information and external links.

Release	Format	Tracks	Country/Date	Label	Catalog#	Barcode	Tagger
Past Masters, Volume One	CD	18	GB 1988-03-07	Parlophone (aka Parlophone UK)	CD-BPM1, CDP 7 90043 2	077779004324	
Past Masters, Volume One	CD	18	US 1988-03-08	Capitol Records (imprint of Capitol Records, Inc.), Parlophone (aka Parlophone UK)	CDP 7 90043 2	077779004324	
Past Masters, Volume One	CD	18	CA 1988	Parlophone (aka Parlophone UK), Apple Records	C2 90043	077779004324	
Past Masters, Volume One	CD	18	CA 1988	Parlophone (aka Parlophone UK)	CDP 7 90043 2	077779004324	
Past Masters, Volume One	CD	18	XE 1988	Parlophone (aka Parlophone UK)	CD-BPM 1, CDP 7 90043 2	077779004324	
Past Masters, Volume One	CD	18	JP 1998-03-11	EMI (double-boxed EMI logo representing Japanese domestic repertoire from 喜芝EMI and EMIミュージック・ジャパン, until the latter's acquisition by Universal Music Japan in	TOCP-51125	4988006740082	

Release group information
Artist: The Beatles
Type: Album + Compilation
Rating: ★★★★ (see all ratings)
Genres: rock
Other tags: 60s
External links: AllMusic, Discogs, Musik-Sammler, Rateyourmusic, Wikidata: Q1473159
Editing: Log in to edit, Open edits

- Un symbole de musique devant un numéro de piste dans le volet de droite indique

qu'aucun fichier n'a été attribué à la piste.

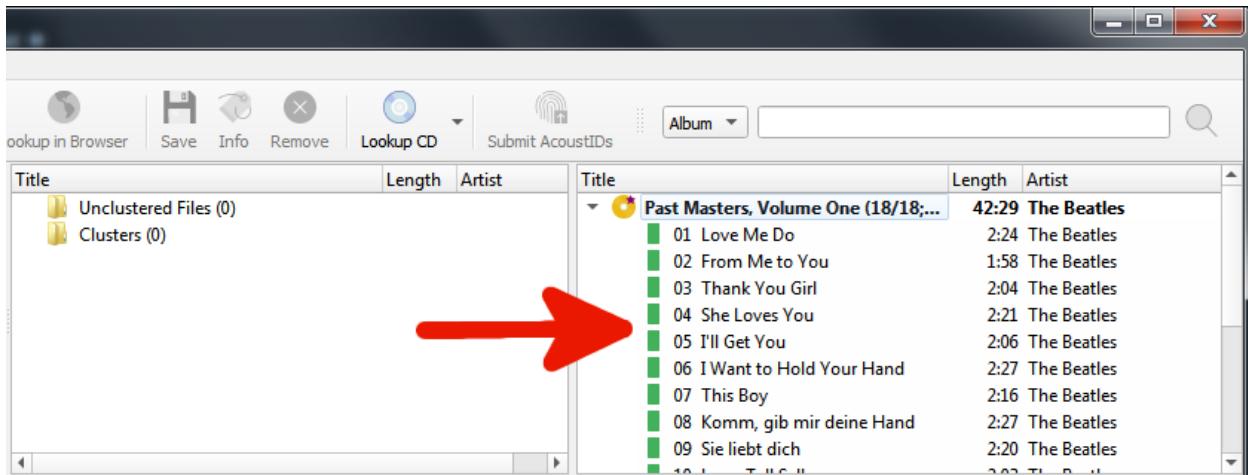


Note: Si vous entrez un lien vers l'entrée souhaitée (par exemple: <https://musicbrainz.org/release/9383a6f5-9607-4a36-9c68-8663aad3592b>) dans le champ de recherche en Picard, l'entrée sera chargée directement sans ouvrir une fenêtre de navigateur.

10.2 Mise en correspondance des fichiers aux pistes

Cette étape est celle où les fichiers individuels sont mis en correspondance avec des pistes spécifiques dans les informations extraites de la base de données MusicBrainz.

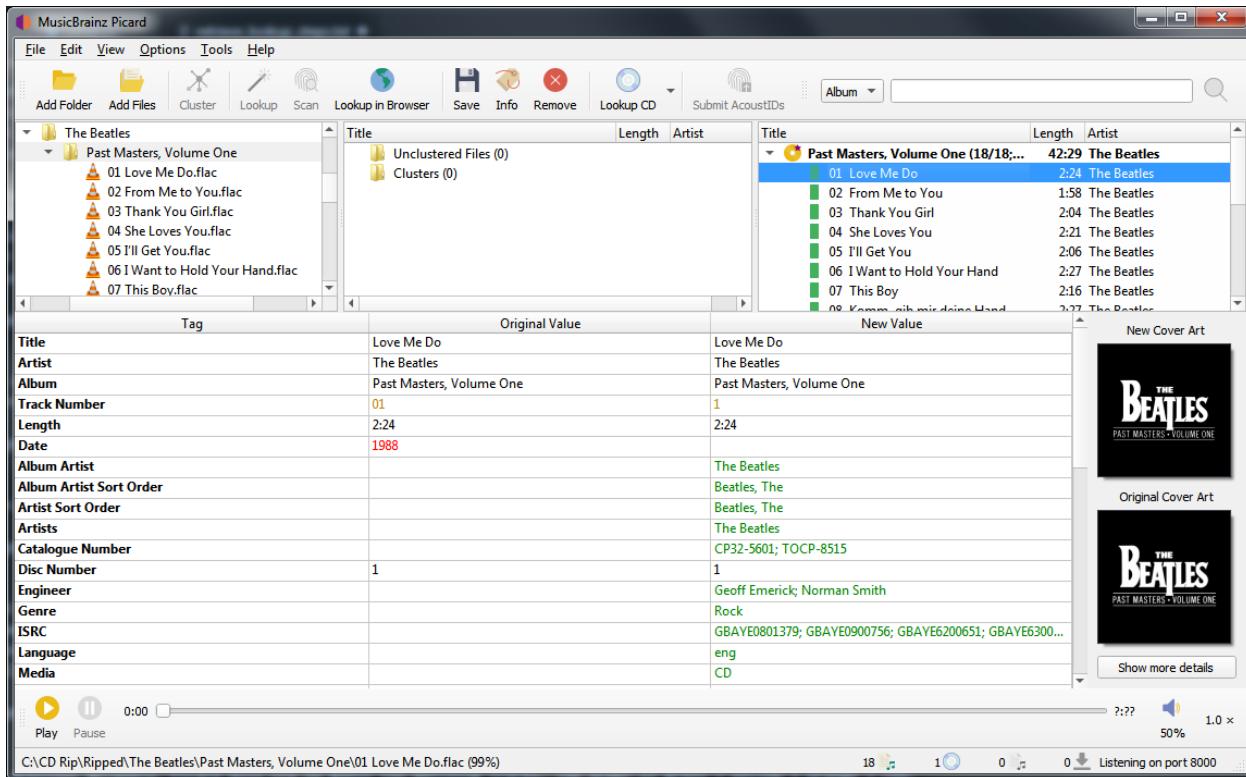
Une fois que vous avez récupéré les informations d'album souhaitées dans le volet de droite, l'étape suivante consiste à faire correspondre les fichiers du volet de gauche à la piste correspondante dans le volet de droite. Un symbole de musique devant un numéro de piste dans le volet de droite indique qu'aucun fichier n'a été attribué à la piste. Dans certains cas, Picard a peut-être déjà essayé de faire l'appariement pour vous. Si la correspondance n'a pas été effectuée automatiquement, faites glisser les fichiers appropriés sur l'album et la piste appropriés.



Note: Si vous glissez et déposez plusieurs fichiers sur une piste spécifique, le premier fichier sélectionné correspondra à la piste sur laquelle vous avez déposé les fichiers. Le reste des fichiers sélectionnés sera mis en correspondance avec les pistes suivantes dans l'ordre. Cela vous permet d'associer rapidement plusieurs fichiers à une séquence de pistes. Si vous voulez faire correspondre tous les fichiers à une seule piste à la place, vous pouvez maintenir la touche Alt tout en déposant les fichiers.

Si vous déposez plusieurs fichiers sur un album, Picard essaiera de faire correspondre les fichiers aux pistes en fonction des métadonnées.

En fonction de vos métadonnées précédentes, Picard essaiera de deviner les pistes correspondantes. L'ordre est vert > jaune > orange > rouge, où le vert correspond le mieux. Si vous voyez beaucoup de rouge et d'orange, cela pourrait signifier que Picard a mal deviné ou que vos fichiers n'avaient pas beaucoup de métadonnées précédentes avec lesquelles travailler. Si tel est le cas, il est recommandé de sélectionner une piste et de comparer les « Valeurs d'origine » et « Nouvelles valeurs » dans le volet des métadonnées. En cas de correspondance incorrecte, faites simplement glisser la piste à son emplacement correct dans le volet de droite.



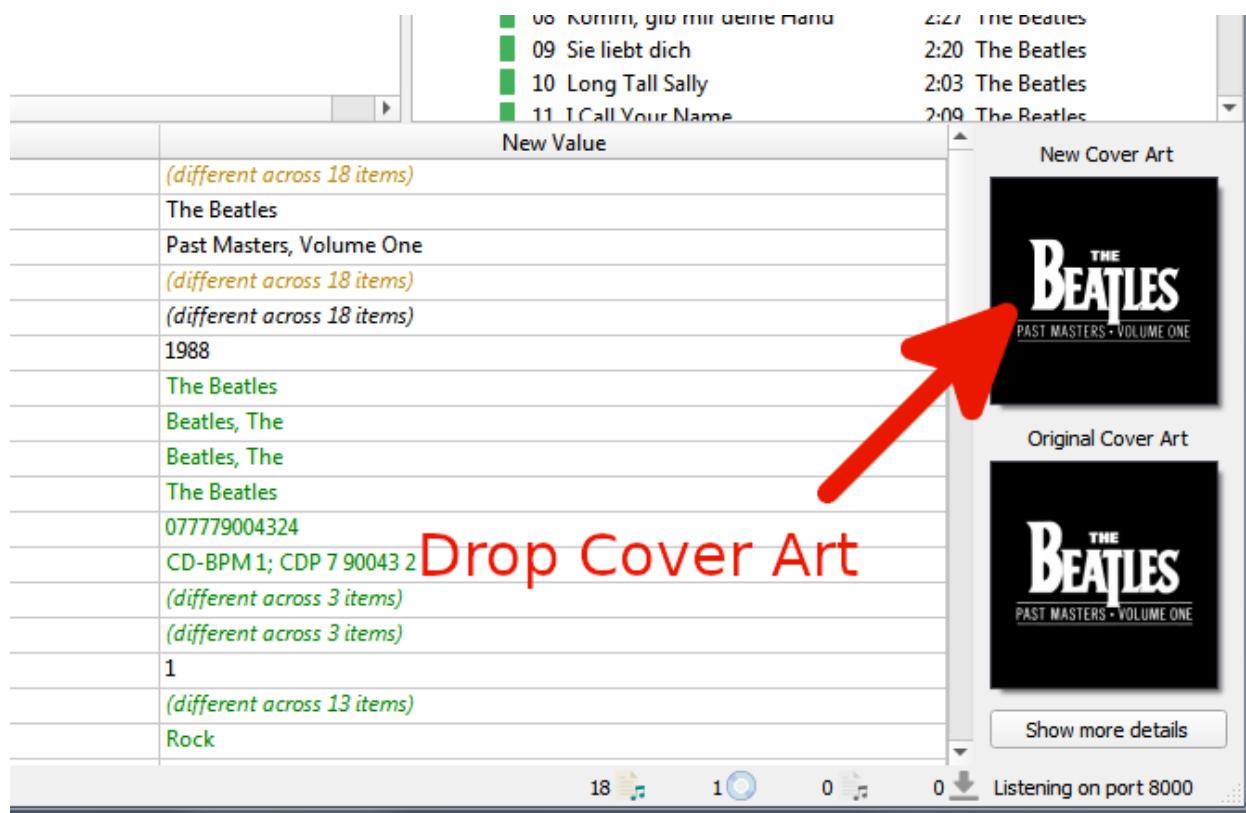
Cliquez avec le bouton droit de la souris sur un élément de la liste des pistes pour afficher un menu de commandes, notamment « Info », « Ouvrir dans le lecteur », « Ouvrir le dossier contenant », « Rechercher des pistes similaires », « Loop dans le navigateur », « Générer des empreintes AcoustID », « Enregistrer » et « Supprimer ». En outre, vous pouvez réexécuter les plugins ou scripts associés uniquement sur l'élément sélectionné. Un clic droit sur un élément dans le volet de gauche fera apparaître un menu similaire de commandes.

Lorsque vous sélectionnez un élément dans le volet de droite, les métadonnées d'origine et les nouvelles de l'élément s'affichent. Un clic droit sur une ligne de la liste des balises affiche un menu de commandes, notamment « Modifier », « Ajouter à la liste "Balises conservées" », « Supprimer » et « Ajouter une nouvelle balise », ainsi qu'une option pour afficher les balises modifiées premier.

10.3 Définition de la pochette

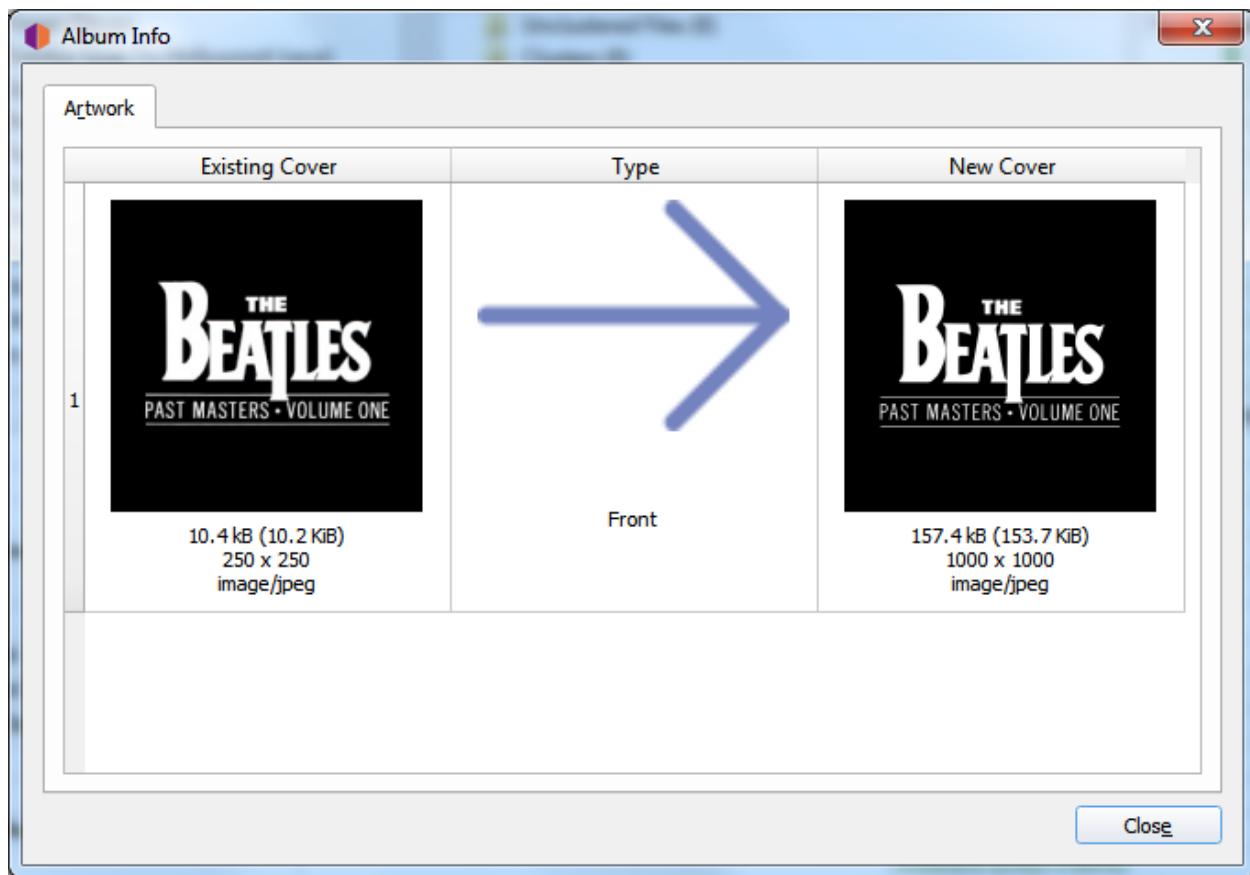
En fonction des paramètres des options, vous pouvez modifier ou confirmer la pochette à enregistrer avec une piste ou un album.

Une fois les informations de sortie téléchargées, la sélection d'un album ou d'une piste dans le volet de droite affichera à la fois la couverture originale et la nouvelle couverture de l'élément sélectionné. Vous pouvez facilement remplacer l'image de couverture utilisée pour l'élément sélectionné en faisant glisser l'image à partir du navigateur de fichiers et en la déposant sur l'image de la nouvelle couverture.



Vous pouvez également choisir un fichier local comme pochette en cliquant avec le bouton droit de la souris sur l'image et en sélectionnant « Choisir un fichier local... » dans le menu.

Le menu propose également des options supplémentaires, notamment « Afficher plus de détails », « Conserver la pochette originale » et des options de traitement des images déposées dans la sélection. Si vous sélectionnez « Afficher plus de détails », une nouvelle fenêtre apparaîtra comme suit:



Un double-clic sur une image ouvrira le fichier image dans le programme par défaut du système pour le type d'image.

10.4 Enregistrement des fichiers mis à jour

Cette étape est celle où Picard met à jour les fichiers correspondants avec les métadonnées récupérées dans la première étape, en fonction des paramètres configurés dans les Options. Cela peut également inclure le changement de nom des fichiers et leur placement dans un répertoire différent.

Lorsque vous êtes convaincu que vos fichiers ont été correctement mis en correspondance avec les pistes dans le volet de droite, sélectionnez l'album que vous souhaitez enregistrer dans le volet de droite et utilisez « *Fichier → Enregistrer* » pour enregistrer le des dossiers. Une coche verte signifie que le fichier a été enregistré à son emplacement approprié.

Title	Length	Artist
• Past Masters, Volume One (18/18;...)	42:28	The Beatles
✓ 01 Love Me Do	2:24	The Beatles
✓ 02 From Me to You	1:58	The Beatles
✓ 03 Thank You Girl	2:04	The Beatles
✓ 04 She Loves You	2:21	The Beatles
✓ 05 I'll Get You	2:06	The Beatles
✓ 06 I Want to Hold Your Hand	2:27	The Beatles
✓ 07 This Boy	2:16	The Beatles
✓ 08 Komm, gib mir deine Hand	2:27	The Beatles
✓ 09 Sie liebt dich	2:20	The Beatles
✓ 10 I ... T H C II	2:07	The Beatles

Une fois que les fichiers ont été sauvegardés avec succès, vous pouvez supprimer l'album du volet de droite en le sélectionnant et en utilisant « *Edit → Remove* ». Notez que cela supprime uniquement l'album de Picard et ne supprime pas les fichiers eux-mêmes.

CHAPTER 11

Recommandations de flux de travail

Cette section fournit quelques flux de travail recommandés pour divers scénarios de balisage. Ces flux de travail sont basés sur ce que l'on pense être les meilleures pratiques.

Les scénarios couverts comprennent:

1. *Lorsque le CD est disponible*
2. *Lorsque les fichiers sont regroupés par album*
3. *Lorsque les fichiers ne sont pas groupés mais ont des métadonnées*
4. *Lorsque les fichiers ne sont pas regroupés et ont peu ou pas de métadonnées existantes*

Note: Qu'il s'agisse ou non de l'un des flux de travail répertoriés, il est **fortement** recommandé de faire une copie de sauvegarde des fichiers en cours de traitement et de traiter initialement une copie de vos fichiers musicaux. Cela aidera à s'assurer que Picard est correctement configuré (par exemple: paramètres, scripts et plugins) et produit les résultats attendus et souhaités.

11.1 Lorsque le CD est disponible

C'est peut-être le meilleur scénario, car il offre les meilleures chances de marquer vos fichiers musicaux avec la correspondance la plus précise de la base de données MusicBrainz. C'est également l'une des méthodes les plus simples pour rechercher la version.

1. Ripper le CD en fichiers musicaux

Extrayez les fichiers musicaux du CD en utilisant votre programme d'extraction préféré (par exemple: [Exact Audio Copy](#) pour Windows ou [Whipper](#) pour Linux). Le format des fichiers de sortie dépend de vos préférences personnelles et des formats pris en charge par votre lecteur. Un format populaire est FLAC, qui est un format compressé sans perte.

2. Recherchez le CD sur MusicBrainz

Avec le CD dans le lecteur, il peut être recherché automatiquement à l'aide de la commande « *Outils → Rechercher un CD* ». Voir la section *CD de recherche* pour des instructions détaillées.

3. Sélectionnez la version appropriée

S'il n'y a qu'une seule version qui correspond à l'ID de disque de votre disque, elle sera chargée automatiquement. Avant de continuer, veuillez vérifier qu'il correspond bien à votre CD (par exemple: pays de sortie, date et étiquette, numéro de catalogue, code-barres, type de support et pochette). Ceci est particulièrement important si vous allez soumettre des informations telles que l'identifiant du disque ou les empreintes digitales AcoustID.

4. Chargez les fichiers

Faites glisser les fichiers ou le dossier du navigateur vers la section « Fichiers non groupés » dans le volet de gauche. Vous n'avez pas besoin de les analyser ou de les regrouper.

5. Faites correspondre les fichiers aux pistes de la version

Faites glisser les fichiers du volet de gauche et déposez-les sur la version dans le volet de droite. Vérifiez que chaque piste de la version est associée à un seul fichier. L'icône de libération devrait devenir dorée. Voir la section *Mise en correspondance des fichiers aux pistes* pour plus de détails.

6. Vérifiez les métadonnées et la pochette

Vérifiez que les métadonnées et l'image de la pochette de la version et des pistes correspondent à vos souhaits. Ajustez si nécessaire. Voir la section *Définition de la pochette* pour plus de détails.

7. Enregistrez les fichiers

Enregistrez les fichiers à l'aide de la commande « *Fichier → Enregistrer* ». Voir la section *Enregistrement des fichiers mis à jour* pour plus de détails.

8. Calculer et soumettre les empreintes digitales AcoustID

Cette étape est facultative, mais appréciée car elle aidera à identifier les fichiers que d'autres pourront rechercher pour le marquage.

Sélectionnez l'entrée de l'album dans le volet de droite et calculez les empreintes AcoustID à l'aide de « *Outils* → *Générer des empreintes AcoustID* ». Une fois les empreintes calculées, soumettez-les en utilisant « *Fichiers* → *Soumettre les AcoustIDs* ».

Note: Les empreintes digitales AcoustID ne doivent être soumises qu'après que les fichiers ont été étiquetés avec les métadonnées MusicBrainz et que vous avez vérifié que les fichiers ont été mis en correspondance avec la bonne piste dans la version appropriée.

11.2 Lorsque les fichiers sont regroupés par album

Si les fichiers musicaux à traiter sont déjà regroupés dans des dossiers par album, le processus de recherche de la version dans la base de données MusicBrainz est grandement simplifié car Picard fonctionne mieux lors du traitement d'un album à la fois.

1. Chargez les fichiers

Faites glisser les fichiers ou le dossier du navigateur vers la section « Fichiers non groupés » dans le volet de gauche.

2. Regrouper et rechercher les fichiers

Sélectionnez les fichiers dans le volet de gauche et combinez-les dans un groupe d'albums à l'aide de la commande « *Outils* → *Cluster* ». Sélectionnez le groupe dans le volet de gauche et lancez la recherche à l'aide de « *Outils* → *Recherche* ». Voir la section *Rechercher des fichiers* pour plus de détails.

3. Sélectionnez la version appropriée

S'il n'y a qu'une seule version qui correspond à la recherche, elle sera chargée automatiquement. Avant de continuer, veuillez vérifier qu'il correspond bien à votre album (par exemple: pays de sortie, date et étiquette, numéro de catalogue, code-barres, type de support et pochette). Ceci est particulièrement important si vous allez soumettre des informations telles que les empreintes digitales AcoustID.

4. Faites correspondre les fichiers aux pistes de la version

Faites glisser les fichiers du volet de gauche et déposez-les sur la version dans le volet de droite. Vérifiez que chaque piste de la version est associée à un seul fichier. L'icône de libération devrait devenir dorée. Voir la section *Mise en correspondance des fichiers aux pistes* pour plus de détails.

5. Vérifiez les métadonnées et la pochette

Vérifiez que les métadonnées et l'image de la pochette de la version et des pistes correspondent à vos souhaits. Ajustez si nécessaire. Voir la section [Définition de la pochette](#) pour plus de détails.

6. Enregistrez les fichiers

Enregistrez les fichiers à l'aide de la commande « *Fichier → Enregistrer* ». Voir la section [Enregistrement des fichiers mis à jour](#) pour plus de détails.

7. Calculer et soumettre les empreintes digitales AcoustID

Cette étape est facultative, mais appréciée car elle aidera à identifier les fichiers que d'autres pourront rechercher pour le marquage.

Sélectionnez l'entrée de l'album dans le volet de droite et calculez les empreintes AcoustID à l'aide de « *Outils → Générer des empreintes AcoustID* ». Une fois les empreintes calculées, soumettez-les en utilisant « *Fichiers → Soumettre les AcoustIDs* ».

Note: Les empreintes digitales AcoustID ne doivent être soumises qu'après que les fichiers ont été étiquetés avec les métadonnées MusicBrainz et que vous avez vérifié que les fichiers ont été mis en correspondance avec la bonne piste dans la version appropriée.

11.3 Lorsque les fichiers ne sont pas groupés mais ont des métadonnées

Dans cette situation, vous devrez envoyer des lots de fichiers à Picard pour les traiter. Afin de minimiser l'impact sur les performances, il est recommandé de garder les lots relativement petits (c'est-à-dire: environ 200 fichiers au maximum dans un seul lot). Picard essaiera de les regrouper en clusters en fonction des métadonnées existant actuellement dans les fichiers.

Note: Ce flux de travail ne fera probablement correspondre que partiellement les fichiers à une version dans chaque lot traité. Cela signifie qu'un album peut ne pas être entièrement mis en correspondance, étiqueté et renommé tant que plusieurs lots n'ont pas été traités.

1. Chargez les fichiers

Faites glisser le lot de fichiers à traiter du navigateur vers la section « Fichiers non groupés » dans le volet de gauche.

2. Regrouper et rechercher les fichiers

Sélectionnez les fichiers dans le volet de gauche et combinez-les en groupes d'albums à l'aide de la commande « *Outils → Cluster* ». Picard tentera de regrouper les fichiers en se basant sur leurs métadonnées existantes. Sélectionnez les groupes souhaités dans le volet gauche et lancez la recherche à l'aide de la commande « *Outils → Recherche* ». Consultez la section *Rechercher des fichiers* pour plus de détails.

3. Faites correspondre les fichiers aux pistes de la version

Faites glisser les fichiers du volet de gauche et déposez-les sur la version dans le volet de droite. Vérifiez que chaque piste de la version est associée à un seul fichier. L'icône de version restera probablement argentée, indiquant que toutes les pistes n'ont pas été associées aux fichiers. Voir la section *Mise en correspondance des fichiers aux pistes* pour plus de détails.

4. Vérifiez les métadonnées et la pochette

Vérifiez que les métadonnées et l'image de la pochette de la version et des pistes correspondent à vos souhaits. Ajustez si nécessaire. Voir la section *Définition de la pochette* pour plus de détails.

5. Enregistrez les fichiers

Enregistrez les fichiers à l'aide de la commande « *Fichier → Enregistrer* ». Voir la section *Enregistrement des fichiers mis à jour* pour plus de détails.

Note: Il n'est pas recommandé de soumettre des empreintes digitales AcoustID pour les fichiers correspondants de cette manière, car il est pratiquement impossible de vérifier que vos fichiers correspondent réellement aux enregistrements correspondants.

11.4 Lorsque les fichiers ne sont pas regroupés et ont peu ou pas de métadonnées existantes

C'est peut-être le pire des cas, car il offre les meilleures chances de marquer vos fichiers musicaux avec une correspondance incorrecte de la base de données MusicBrainz.

Dans cette situation, vous devrez envoyer des lots de fichiers à Picard pour les traiter. Afin de minimiser l'impact sur les performances, il est recommandé de garder les lots relativement petits (c'est-à-dire: environ 200 fichiers au maximum dans un seul lot). Picard essaiera de les regrouper en groupes en fonction de leurs empreintes digitales AcoustID.

Note: Ce flux de travail ne fera probablement correspondre que partiellement les fichiers à une version dans chaque lot traité. Cela signifie qu'un album peut ne pas

être entièrement mis en correspondance, étiqueté et renommé tant que plusieurs lots n'ont pas été traités.

1. Chargez les fichiers

Faites glisser le lot de fichiers à traiter du navigateur vers la section « Fichiers non groupés » dans le volet de gauche.

2. Analyser les fichiers

Sélectionnez les fichiers dans le panneau de gauche et scannez-les en utilisant la commande « *Outils* → *Analyser* ». Picard essaiera de calculer l'empreinte AcoustID pour chacun des fichiers et ensuite de récupérer les communiqués avec les enregistrements correspondants. Voir la section *Analyser les fichiers* pour plus de détails.

3. Faites correspondre les fichiers aux pistes de la version

Faites glisser les fichiers du volet de gauche et déposez-les sur la version dans le volet de droite. Vérifiez que chaque piste de la version est associée à un seul fichier. L'icône de version restera probablement argentée, indiquant que toutes les pistes n'ont pas été associées aux fichiers. Voir la section *Mise en correspondance des fichiers aux pistes* pour plus de détails.

4. Vérifiez les métadonnées et la pochette

Vérifiez que les métadonnées et l'image de la pochette de la version et des pistes correspondent à vos souhaits. Ajustez si nécessaire. Voir la section *Définition de la pochette* pour plus de détails.

5. Enregistrez les fichiers

Enregistrez les fichiers à l'aide de la commande « *Fichier* → *Enregistrer* ». Voir la section *Enregistrement des fichiers mis à jour* pour plus de détails.

CHAPTER 12

Autres tâches de Picard

En plus de la fonctionnalité générale de balisage et d'organisation de vos fichiers audio Picard, il existe d'autres tâches qu'il peut effectuer:

12.1 Joindre un ID de disque à une version

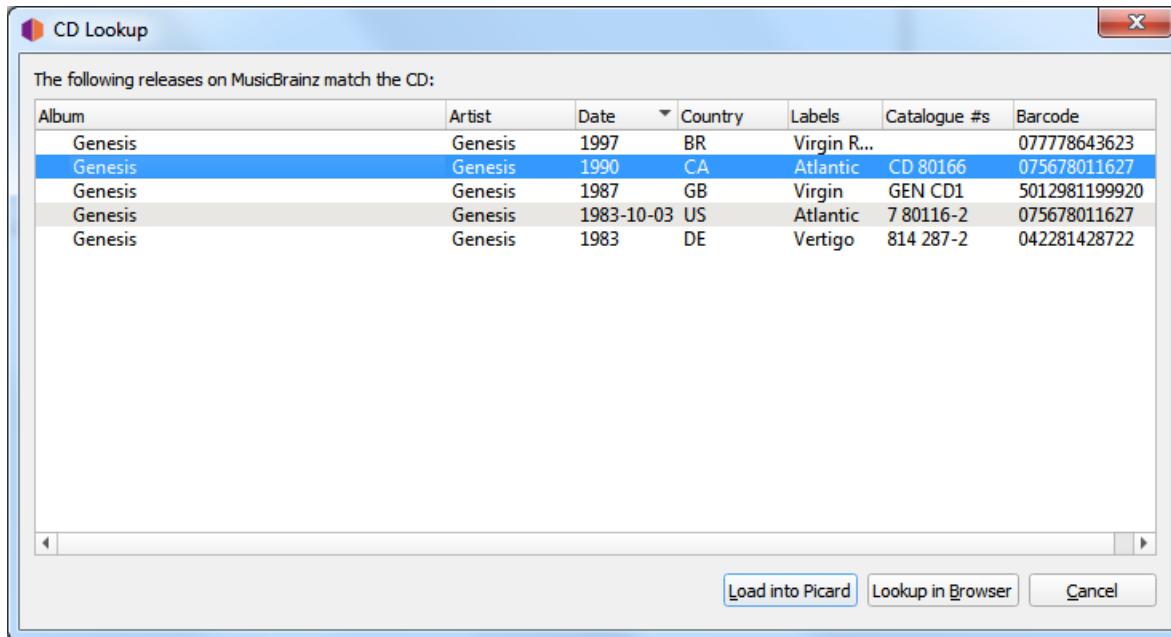
Les ID de disque sont très utiles pour identifier les CD et permettre à MusicBrainz de connaître la longueur des pistes d'un CD. Ainsi, il est très utile de les ajouter lors de la soumission d'une nouvelle version de CD ou lorsque vous avez une version de CD qui n'a pas d'identifiant de disque attaché.

Note: Veuillez ne pas ajouter de DisclD à partir de CD gravés à la maison.

Les étapes à suivre pour soumettre un identifiant de disque sont:

1. Assurez-vous que le CD est inséré dans le lecteur et sélectionnez « *Outils → Rechercher un CD... → (lecteur à utiliser)* ». Le toc du CD sera calculé et envoyé à MusicBrainz, et une liste des versions correspondantes sera affichée.

MusicBrainz Picard, Version v2.6.3



- Si la version cible apparaît dans cette liste, l'ID du disque a déjà été joint et vous n'avez rien d'autre à faire. Si aucune correspondance n'est trouvée ou si la version cible souhaitée n'apparaît pas dans la liste, utilisez l'option « Soumettre l'ID du disque » pour localiser la version correcte. Entrez le titre de la sortie ou l'artiste et lancez la recherche. Vous recevrez une liste des sorties qui correspondent à votre critère de recherche et qui ont le même nombre de pistes que votre CD toc.

Position	Title	Artist	Format	Country/Date	Label	Catalog#	Barcode	Tagger
1/1	Genesis	Genesis	CD	CA 1990	Atlantic (Warner Music imprint)	CD 80166	075678011627	
1/1	Genesis	Genesis	CD	BR 1997	Virgin Records America (DO NOT USE! please refer to either "Virgin" or "Virgin America")		077778643623	
1/1	Genesis	Genesis	CD	US 1983-10-03	Atlantic (Warner Music imprint)	7 80116-2	075678011627	
1/1	Genesis	Genesis	CD	GB 1987	Virgin (worldwide imprint of Virgin Records Ltd. and all its subsidiaries)	GEN CD1	5012981199920	
1/1	Genesis	Genesis	CD	DE 1983	Vertigo (British rock label)	814 287-2	042281428722	

We used DiscID 31xFlVam0kxDmvnAjXtvX4b91s- to look up this information.

Search by artist:

Search by release:

[Donate](#) | [Wiki](#) | [Forums](#) | [IRC](#) | [Bug Tracker](#) | [Blog](#) | [Twitter](#) | [Use beta site](#)

Brought to you by [MetaBrainz Foundation](#) and our [sponsors](#) and [supporters](#). Cover Art provided by the [Cover Art Archive](#).

Note: Si vous recherchez par artiste, utilisez le bouton radio à côté du nom de l'artiste pour sélectionner l'artiste souhaité. Si vous cliquez sur le lien du nom de l'artiste, vous ne serez pas autorisé à joindre l'identifiant du disque à l'une des versions affichées.

3. Sélectionnez la version cible souhaitée dans la liste affichée en cliquant sur le bouton radio à côté de la version, puis cliquez sur le bouton « Joindre le CD TOC » sous la liste des versions. Cela préparera une édition pour joindre l'identifiant du disque à la version. Vous devez ensuite ajouter une note d'édition appropriée et soumettre l'édition.

Release Group: Splendid Genesis		Release Group: Genesis	
Splendid Genesis	Abstract Reason	UK XW	2015-01-24
<input type="radio"/> Digital Media 1 (show tracklist)			[none]
Genesis	Genesis	GB	1983
<input type="radio"/> Cassette 1 (show tracklist)	Charisma		
Genesis	Genesis	DE	1983
<input type="radio"/> CD 1 (show tracklist)	Vertigo (British rock label)		
Genesis	Genesis	US	1983-10-03
<input type="radio"/> Digital Media 1 (show tracklist)	Atlantic (Warner Music imprint)		
Genesis	Genesis	XE	-
<input type="radio"/> CD 1 (show tracklist)	Charisma, Virgin (worldwide imprint of Virgin Records Ltd. and all its subsidiaries) 7 86436 2, GEN CD1		0077778643623
Genesis	Genesis	JP	1985-06-01
<input type="radio"/> CD 1 (show tracklist)	Vertigo (British rock label)		32PD-17 4988011303999
Genesis	Genesis	NL	1983-10-03
<input type="radio"/> 12" Vinyl 1 (show tracklist)	Mercury Records (or just "Mercury". A UMG imprint, do not use it for ©/© credits)		832 178-1 042281428715
Genesis	Genesis	CA	1990
<input type="radio"/> CD 1 (show tracklist)	Atlantic (Warner Music imprint)		CD 80166 075678011627
Genesis made in Japan	Genesis	XE	-
<input type="radio"/> CD 1 (show tracklist)	Atlantic (Warner Music imprint)		7 80116-2, 814287-2 [none]
CD 1 (show tracklist)			

4. Si aucune des versions affichées n'est correcte, vous avez la possibilité d'ajouter une nouvelle version (avec certaines informations automatiquement incluses). L'identifiant du disque sera automatiquement attaché à la nouvelle version lorsque l'édition est enregistrée.

CD 1 (show tracklist)

Attach CD TOC

Add a new release

If you don't see the release you are looking for, you can still add a new one, using this CD TOC:

Add a new release

Donate | Wiki | Forums | IRC | Bug Tracker | Blog | Twitter | Use beta site

12.2 Soumission d'empreintes acoustiques

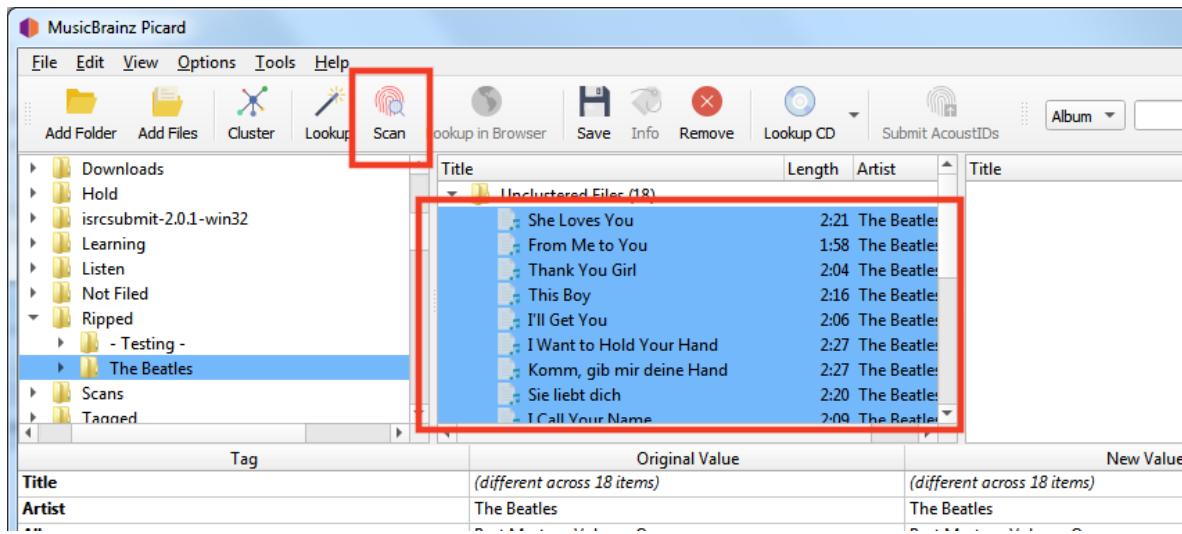
Les empreintes acoustiques sont très utiles pour identifier les pistes et les enregistrements, ce qui permet de les rechercher dans la base de données MusicBrainz. Ainsi, il est très utile de les ajouter lorsque vous étiquetez des fichiers. Veuillez consulter le didacticiel [Compréhension Empreintes acoustiques et AcoustIDs](#) pour plus d'informations.

Note: Lorsque vous utilisez Picard pour soumettre des empreintes acoustiques, il est recommandé d'activer la colonne Empreintes digitales dans la vue du tableau dans le volet de droite. Pour ce faire, cliquez avec le bouton droit de la souris sur l'en-tête de la colonne et cochez la case « État de l'empreinte digitale ». Cela affichera une icône indiquant si l'AcoustID a été calculé et s'il est prêt à être soumis (rouge = non soumis, gris = déjà soumis).

Il existe deux méthodes pour soumettre des empreintes acoustiques, en fonction du flux de travail que vous utilisez pour identifier les versions que vous marquez. Les étapes à suivre pour soumettre des empreintes acoustiques pour chacun des deux flux de travail sont les suivantes :

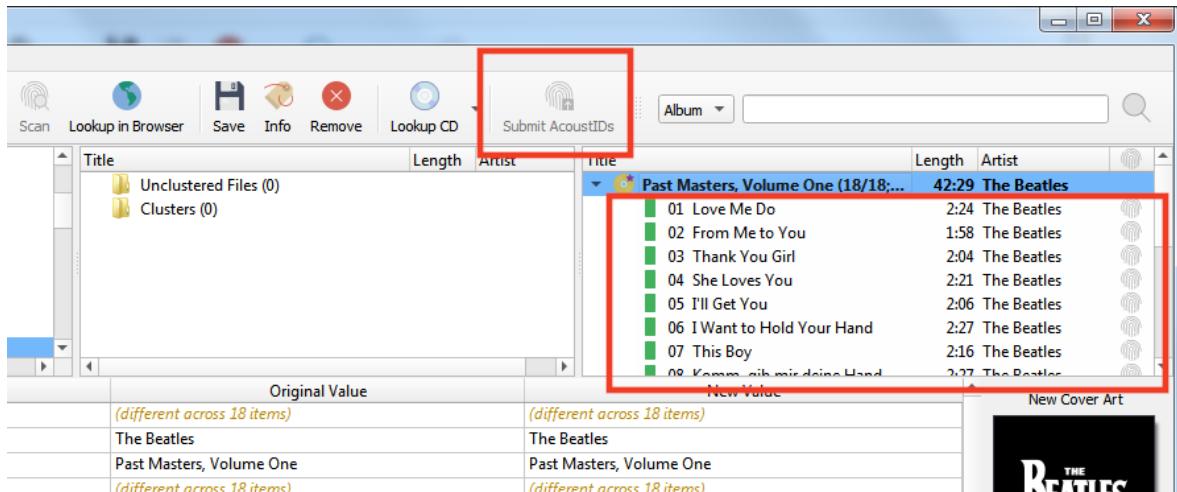
12.2.1 Envoi lors de l'utilisation de Scan pour identifier la version

1. Chargez les fichiers dans le volet de clustering. Sélectionnez les fichiers et cliquez sur le bouton « Scan », ou sélectionnez « Outils → Scan ».

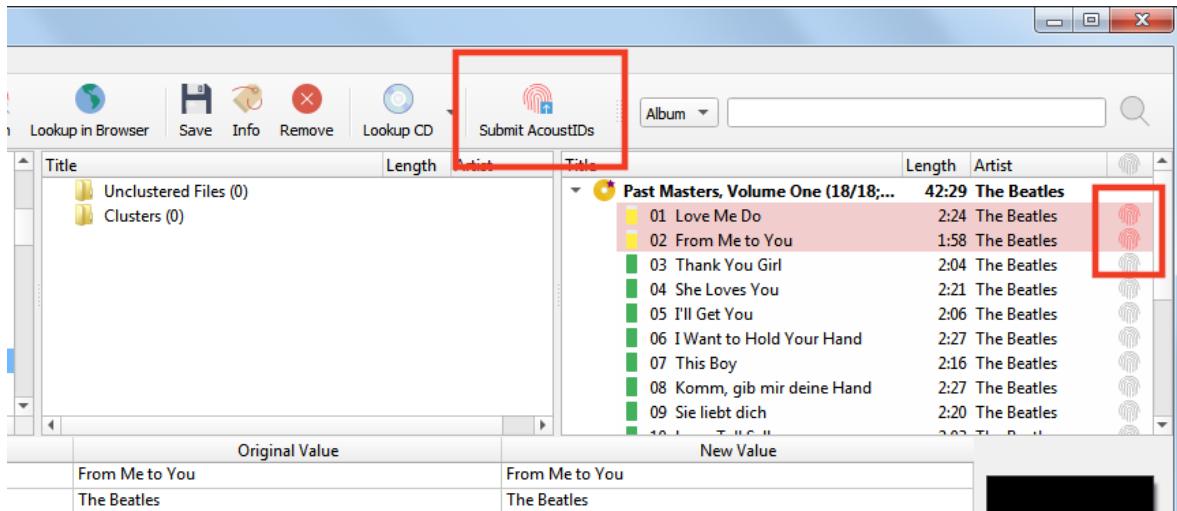


2. Si les fichiers correspondent à une piste et sont déplacés vers le volet de droite, ils

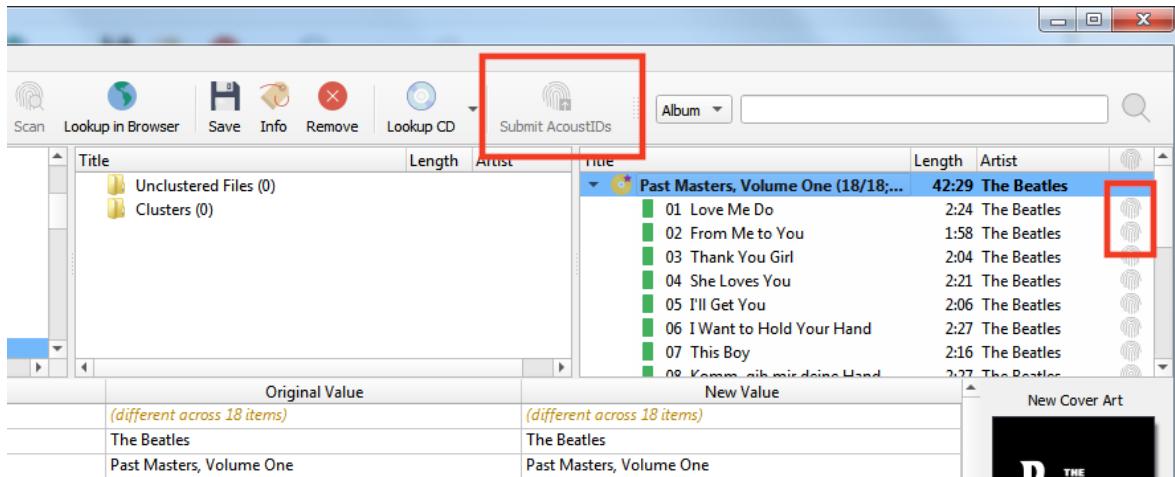
existent déjà dans la base de données AcoustID et n'ont pas besoin d'être soumis à nouveau. Le bouton « Soumettre » restera désactivé.



- Si les fichiers ne correspondent pas ou si vous les déplacez manuellement pour qu'ils correspondent à une autre piste, ils pourraient être soumis. L'icône AcoustID pour les pistes apparaîtra en rouge (c'est-à-dire: statut non soumis) et le bouton « Soumettre » sera activé.

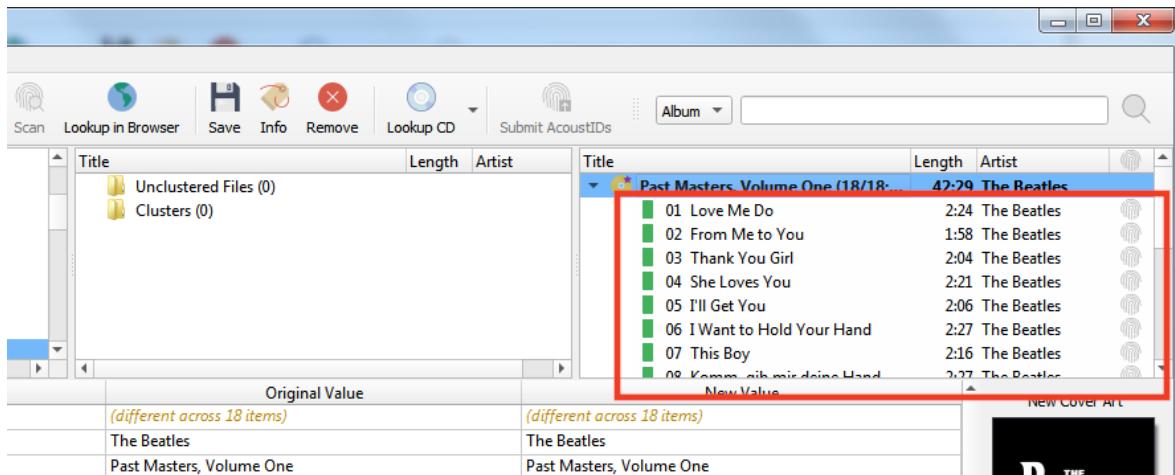


- Cliquer sur le bouton « Soumettre » ne soumettra que les empreintes digitales des fichiers identifiés à l'étape 3. L'icône AcoustID des pistes deviendra grise (c'est-à-dire: statut soumis) et le bouton « Soumettre » sera désactivé.

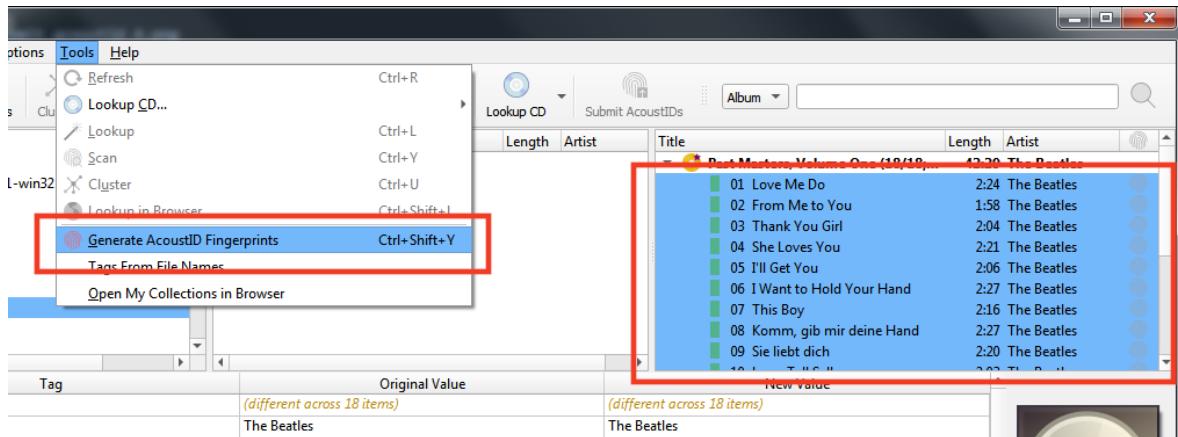


12.2.2 Soumission lorsque vous n'utilisez pas Scan pour identifier la version

- Assurez-vous que les fichiers correspondent correctement aux pistes d'une version dans le volet de droite.

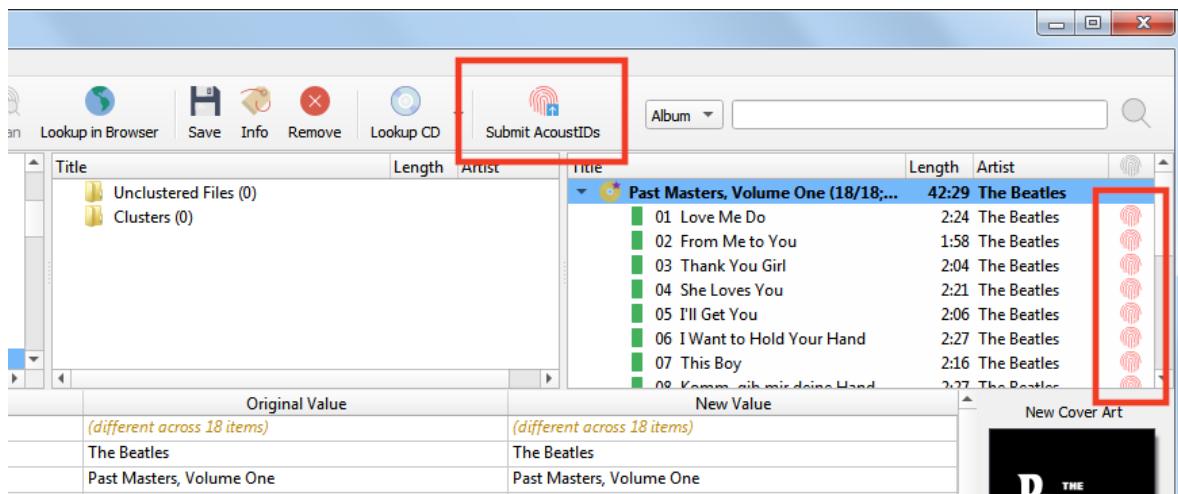


- Sélectionnez les fichiers dans le volet de droite et sélectionnez « Outils → Générer des empreintes AcoustID ». Cela calculera les empreintes acoustiques des fichiers sélectionnés.

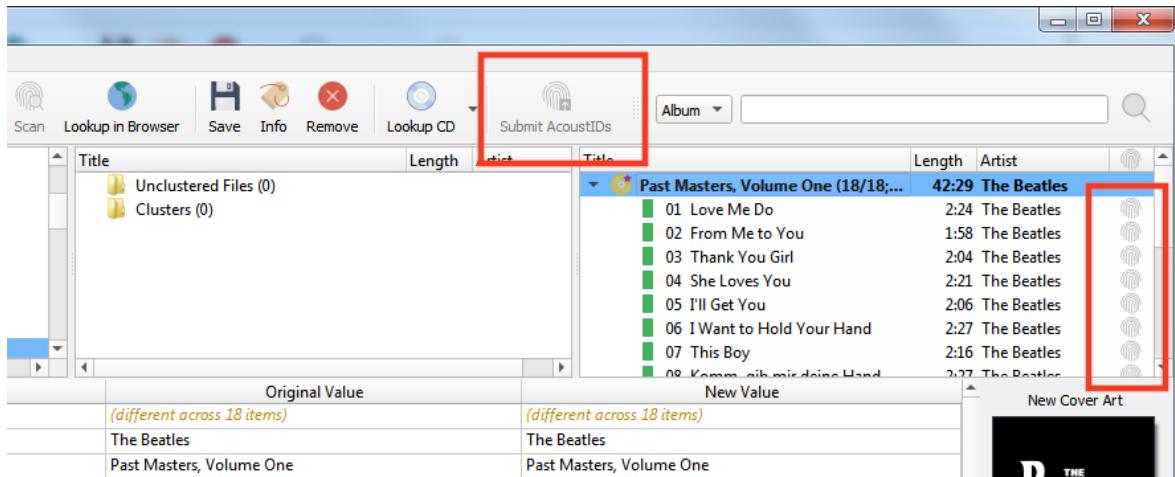


Note: Le bouton d'action « Générer des empreintes digitales AcoustID » peut être ajouté à la barre de boutons en modifiant les paramètres dans les options de l'interface utilisateur.

L'icône AcoustID pour les pistes apparaîtra en rouge (c'est-à-dire: statut non soumis) et le bouton « Soumettre » sera activé.



3. Cliquer sur le bouton « Soumettre » soumettra les empreintes digitales des fichiers. L'icône AcoustID des pistes deviendra grise (ex: état soumis) et le bouton « Soumettre » sera désactivé.

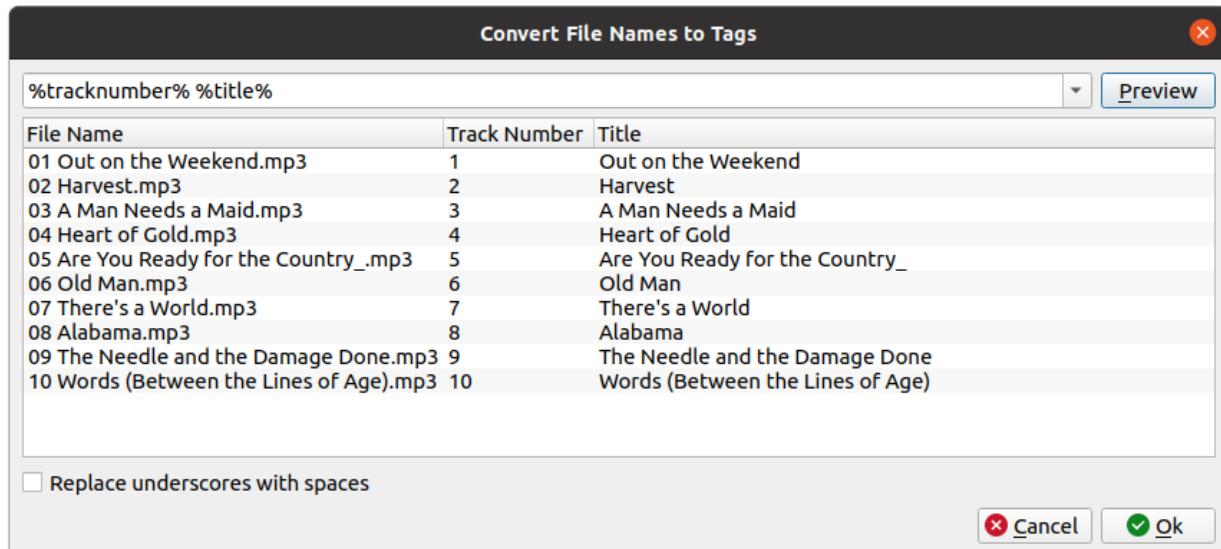


12.3 Générer des balises à partir des noms de fichiers

Parfois, les fichiers ont des balises de mauvaise qualité ou aucune balise du tout, mais les noms de fichiers sont bien structurés et suivent un modèle. Dans ce cas, vous pouvez utiliser « *Outils → Balises de noms de fichiers...* » pour générer les balises à partir des noms de fichiers.

12.3.1 Utilisation de base

Pour utiliser cet outil, sélectionnez un ou plusieurs fichiers chargés dans Picard et ouvrez la boîte de dialogue *Balises à partir de noms de fichiers* dans le menu sous « *Outils → Balises à partir de noms de fichiers...* ». La boîte de dialogue vous montrera une liste de noms de fichiers et un champ de saisie en haut où vous pouvez entrer un modèle correspondant.



Le modèle de correspondance peut être constitué de noms de balises Picard entourés de signes % et d'autres caractères correspondant textuellement. Pour les noms de balises, vous pouvez utiliser des noms prédéfinis tels que %artist%, %album% ou %title% (voir [Balises & Variables](#)) ou utilisez des noms personnalisés. Vous pouvez sélectionner quelques modèles prédéfinis, mais vous pouvez également les ajuster ou définir les vôtres.

Si vos fichiers se composent par exemple d'un numéro de piste et d'un titre de piste séparés par un espace (par exemple 04 Heart of Gold.mp3), vous pouvez utiliser le modèle de correspondance %tracknumber% %title%. Si le numéro de la piste et le titre sont séparés par un trait d'union, comme 04 - Heart of Gold.mp3, le modèle doit également inclure ce séparateur, comme %tracknumber% - %title%.

Cliquez sur le bouton « Aperçu » à côté du modèle correspondant pour afficher un aperçu des balises extraites pour chaque nom de fichier. Une fois que vous êtes satisfait du résultat, vous pouvez accepter les modifications avec le bouton « Ok ». Les balises modifiées seront définies pour les fichiers. Notez que les modifications ne seront pas enregistrées automatiquement, vous devez tout de même enregistrer les fichiers si vous souhaitez que les balises soient écrites (voir [Enregistrement des fichiers mis à jour](#)).

12.3.2 Dossiers correspondants

Le modèle peut également correspondre aux dossiers parents du fichier. Pour faire correspondre les dossiers, utilisez une barre oblique (/) comme séparateur. Si par exemple le fichier se trouve dans un dossier nommé d'après l'album, qui à son tour se trouve dans un dossier nommé d'après l'artiste (ex Neil Young/Harvest/04 Heart of Gold.mp3), vous pouvez faire correspondre l'artiste, album, numéro de piste et titre avec un modèle de %artist%/%album%/%tracknumber% - %title%.

File Name	Artist	Album	Track Number	Title
01 Out on the Weekend.mp3	Neil Young	Harvest	1	Out on the Weekend
02 Harvest.mp3	Neil Young	Harvest	2	Harvest
03 A Man Needs a Maid.mp3	Neil Young	Harvest	3	A Man Needs a Maid
04 Heart of Gold.mp3	Neil Young	Harvest	4	Heart of Gold
05 Are You Ready for the Country_.mp3	Neil Young	Harvest	5	Are You Ready for the Country_
06 Old Man.mp3	Neil Young	Harvest	6	Old Man
07 There's a World.mp3	Neil Young	Harvest	7	There's a World
08 Alabama.mp3	Neil Young	Harvest	8	Alabama
09 The Needle and the Damage Done.mp3	Neil Young	Harvest	9	The Needle and the Damage Done
10 Words (Between the Lines of Age).mp3	Neil Young	Harvest	10	Words (Between the Lines of Age)

Replace underscores with spaces

Cancel Ok

12.3.3 Remplacer les traits de soulignement par des espaces

Parfois, les fichiers ont été nommés sans espaces et utilisent plutôt des traits de soulignement. Par exemple, un fichier pourrait être nommé 04_Heart_of_Gold.mp3. Par défaut, le titre serait extrait en tant que « Heart_of_Gold ». Dans ce cas, cochez la case « Remplacer les traits de soulignement par des espaces » et utilisez un modèle comme %tracknumber% _%title% pour extraire correctement le titre en « Heart of Gold ».

12.3.4 Ignorer certaines parties du nom de fichier

Parfois, vous ne souhaitez pas inclure des parties du nom de fichier dans vos balises et souhaitez simplement les ignorer. Le modèle doit toujours correspondre au nom de fichier entier, cependant. Dans ce cas, vous pouvez utiliser une variable masquée pour les parties du nom de fichier que vous ne souhaitez pas faire correspondre à une balise réelle. Les variables masquées commencent par un trait de soulignement comme %_dummy%. Cette variable sera toujours disponible dans le fichier pour *Script*, mais ne sera pas écrite dans les balises réelles du fichier lors de l'enregistrement.

Un exemple pourrait être si vous souhaitez extraire uniquement le numéro de piste d'un nom de fichier comme 04 Are You Ready for the Country_.mp3. Peut-être que la balise de numéro de piste est manquante dans le fichier, mais la balise de titre est déjà correctement définie. Vous pouvez utiliser le modèle %tracknumber% %_filetitle%. Cela extraire correctement la balise tracknumber, mais extraire le reste du nom de fichier dans une variable cachée %_filetitle% qui ne sera pas écrite dans les balises de fichier. Le nom %_filetitle% est choisi arbitrairement, il suffit de commencer par un trait de soulignement.

Note: L'analyse des variables cachées à partir des noms de fichiers est prise en charge depuis Picard 2.5. Les versions antérieures créeraient une balise réelle qui serait stockée dans les balises. Si vous utilisez une version Picard antérieure à 2.5, vous devrez supprimer les balises indésirables avant d'enregistrer les fichiers.

CHAPTER 13

Prolonger Picard

Il existe deux manières principales d'étendre la fonctionnalité de MusicBrainz Picard: *plugins* et *scripts*.

Les plugins peuvent être installés / désinstallés et activés / désactivés à partir du menu Options. Les plugins installés sont chargés lors du démarrage de Picard, et sont mis à disposition du programme.

Les scripts sont stockés dans les paramètres utilisateur, et sont gérés à partir du menu « *Options → Options...* ».

13.1 Plugins

Les plugins sont écrits en Python et sont enregistrés dans les hooks appropriés. Chaque plugin a son propre identifiant de version, mais répertorie également les versions d'API de plugin qu'il prend en charge. Lors du chargement d'un plugin, Picard compare d'abord sa liste de versions d'API aux versions prises en charge par le plugin pour s'assurer que le plugin fonctionnera correctement. Les versions de l'API Picard indiquent la version du programme dans laquelle l'API du plugin a été mise à jour pour la dernière fois et toutes les API du plugin avec lesquelles elle est rétrocompatible.

Les hooks sont des connexions aux différents objets de Picard qui appellent un type spécifique de plugin. Pendant le fonctionnement normal de Picard, lorsqu'il rencontre un hook, il récupère d'abord une liste de tous les plugins enregistrés pour ce hook spécifique, puis les exécute séquentiellement dans l'ordre en fonction de la priorité spécifiée lors de l'enregistrement du plugin sur le hook.

Il existe différents types de plugins, notamment:

Processeurs de métadonnées: Ces plugins peuvent accéder et modifier les métadonnées lors de leur chargement depuis MusicBrainz. Ils sont enregistrés avec `register_album_metadata_processor()` ou `register_track_metadata_processor()`. Ce sont ce que vous pourriez appeler « automatiques » car ils fonctionnent sans aucune intervention de l'utilisateur. Un exemple est le plugin Classical Extras.

Fournisseurs d'art de couverture: Ces plugins fournissent une autre source d'art de couverture et sont enregistrés avec `register_cover_art_provider()`. Ils sont également « automatiques » en ce sens qu'ils chargent la pochette d'album sans intervention de l'utilisateur, bien qu'ils doivent être activés par l'utilisateur dans les options de la pochette. Le plugin Fanart.tv en est un exemple.

Fonction de script: Certains plugins fournissent simplement des fonctions de script supplémentaires à utiliser dans « *Options → Scripting* » ou le script de renommage. Ceux-ci sont enregistrés avec `register_script_function()`. Keep tag, qui fournit la fonction `$keep()`, en est un exemple.

Actions du menu contextuel: les plugins peuvent enregistrer des actions qui peuvent être activées manuellement via le menu contextuel. C'est ce que fait le plugin Load as non-album track. Un autre exemple est Generate CueSheet. Ceux-ci sont enregistrés avec `register_album_action()`, `register_track_action()`, `register_file_action()`, `register_cluster_action()` ou `register_clusterlist_action()`.

Formats de fichiers: les plugins peuvent également prendre en charge de nouveaux formats de fichiers non encore pris en charge par Picard. Ceux-ci sont enregistrés avec `register_format()`.

Processeurs d'événements: les plugins peuvent s'exécuter automatiquement en fonction de certains déclencheurs d'événements. Ceux-ci sont enregistrés avec `file_post_load_processor()`, `file_post_save_processor()`, `file_post_addition_to_track_processor()`, `file_post_removal_from_track_processor()` ou `album_post_removal_processor()`.

Notez que les plugins ne sont pas limités à l'un de ces domaines. Un seul plugin pourrait implémenter tout ce qui précède, mais la plupart des plugins existants se concentrent sur un seul.

L'API [Plugins API](#) fournit des informations sur les différents plugins hooks disponibles, ainsi que quelques exemples de leur utilisation. Il existe également une liste des [plugins disponibles](#) qui ont été soumis au référentiel MusicBrainz Picard affiché sur le site Picard.

13.2 Scripts

Il existe deux types de scripts utilisés dans Picard: le script de nommage des fichiers et les scripts de marquage. Ils sont gérés à partir des sections « Nommage des fichiers » et « Scripting » du menu « Options → Options... ». Tous les scripts sont écrits en utilisant le [Langage de script Picard](#). Les scripts sont souvent discutés dans le [Forum communautaire de MetaBrainz](#), et il y a un fil spécifique au [nommage des fichiers](#) et aux extraits de scripts.

13.2.1 Script de dénomination de fichier

Il n'y a qu'un seul script de dénomination de fichier défini dans les paramètres d'un utilisateur, bien qu'il puisse varier d'un simple script sur une ligne tel que `%album%/%title%` à un script très complexe utilisant différents formats de dénomination de fichier basés sur différents Critères. Dans tous les cas, les fichiers seront enregistrés en utilisant le texte sorti par le script.

Note: Les nouveaux ensembles de balises ou les balises modifiées par le script de dénomination de fichier ne seront pas écrites dans les métadonnées des fichiers de sortie.

13.2.2 Scripts de balisage

Plusieurs scripts de balisage peuvent être définis dans les paramètres d'un utilisateur. Des scripts individuels peuvent être activés ou désactivés, et l'ordre d'exécution des scripts peut être défini. Chaque fois qu'un script est exécuté automatiquement (c'est-à-dire lorsqu'un album est chargé), il est traité une fois pour chaque piste de l'album qui a déclenché la lecture. Par exemple, si deux scripts de balisage sont activés (A et B) et qu'un album avec trois pistes est chargé, les scripts seront traités dans l'ordre suivant:

1. Script A Track 1;
2. Script A Track 2;
3. Script A Track 3;
4. Script B Track 1;
5. Script B Track 2;
6. Script B Track 3.

Les mises à jour des métadonnées ne sont pas partagées entre les pistes, vous ne pouvez donc pas ajouter de données d'une piste à une balise dans une autre piste.

Tout nouveau jeu de balises ou balises modifiées par les scripts de balisage sera écrit dans les métadonnées des fichiers de sortie, sauf si le nom de balise commence par un

trait de soulignement. Ces balises « masquées » sont généralement utilisées comme variables pour contenir des valeurs temporaires qui seront utilisées ultérieurement dans les scripts de balisage ou de dénomination de fichier. Les scripts de balisage sont exécutés une fois pour chaque piste dans les données, en utilisant les métadonnées de cette piste.

Les scripts de balisage peuvent également être exécutés manuellement en cliquant avec le bouton droit sur un album ou une piste dans le volet de droite de Picard. S'il est exécuté à partir de l'entrée de l'album, le script est exécuté pour chaque piste de l'album. S'il est exécuté à partir d'une piste individuelle, le script n'est exécuté que pour cette piste.

13.2.3 Exemples de scripts de marquage

Les exemples de script suivants montrent comment les scripts du tagger peuvent être utilisés pour résoudre certains cas d'utilisation spécifiques. Veuillez vous référer à *Langage de script Picard* pour une description détaillée des variables et des fonctions utilisées dans ces exemples.

Déplacer la désambiguisation vers le titre de l'album

Ajoute le commentaire de désambiguisation d'une sortie au titre de l'album:

```
$set(album,%album%$if(%_releasecomment%, \(%_releasecomment%\)))
```

Libérer la langue en tant que langue

La variable %_releaselanguage% spécifie la langue de la liste des pistes, alors que la variable %language% est supposée être la langue des paroles. Le script suivant utilisera la variable %_releaselanguage% à la place si %language% est vide:

```
$if($not(%language%),$set(language,%_releaselanguage%))
```

Utiliser la date de sortie originale

Par défaut, Picard fournit un tag date qui contient la date de sortie d'une version spécifique et originaldate qui fournit la date de sortie la plus ancienne de cette version. Par exemple, vous pouvez avoir une réédition en 2020 d'un album qui est sorti en 1992. Dans ce cas, date sera mis à « 2020 » et originaldate à « 1992 ». Si vous préférez avoir toujours la date de sortie originale comme date principale dans les balises de votre fichier, vous pouvez utiliser le script suivant:

```
$set(date,$if2(%originaldate%,%date%))
```

L'utilisation de `$if2` garantit que si `originaldate` est vide, il reviendra à date.

De plus, Picard fournit une variable `%_recording_firstreleasedate%`, qui essaie de fournir la première date de sortie par enregistrement (qui peut être différente pour chaque piste dans une sortie). Si vous préférez cela, vous pouvez utiliser le script suivant:

```
$set(date,$if2(%_recording_firstreleasedate%,%originaldate%,%date%))
```

Ou si vous voulez garder le date pour la date de sortie réelle de la version spécifique, mais utiliser la date de première sortie de l'enregistrement comme `originaldate`:

```
$set(originaldate,$if2(%_recording_firstreleasedate%,%originaldate%))
```

Définir le nom de tri de l'album

La balise `albumsrt` n'est pas remplie par Picard par défaut. Vous pouvez lui donner une valeur significative avec les préfixes « The » et « A » déplacés à la fin avec le script suivant:

```
$set(albumsort,$swapprefix(%album%))
```

Par exemple, le nom de tri de la version « The Best of Muddy Waters » deviendra « Best of Muddy Waters, The ».

Définir la compilation pour les sorties multi-artistes

Par défaut, la balise `compilation` sera mise à 1 uniquement pour les sorties `Various Artists`. Le script suivant le définira pour toutes les sorties avec plus d'un artiste (comme c'était le comportement par défaut dans Picard 1.2 et antérieur):

```
$if(%_multiartist%,$set(compilation,1))
```

Supprimer le featuring de l'artiste de l'album

Cela supprime toujours les artistes en vedette de l'artiste de l'album:

```
$set(albumartist,$rreplace(%albumartist%,\\s+feat\\.*,,))
```

Déplacer le featuring de l'artiste au titre

Selon les directives de MusicBrainz, les artistes en featuring font partie du nom de l'artiste, par exemple « Artist A feat. Artist B ». Certains utilisateurs préfèrent que le featuring soit ajouté au titre de l'album ou de la piste. Le script suivant déplace les artistes en vedette vers le titre de la piste:

```
$set(_feat_title,$rsearch(%artist%,\\s+\\\\(?\\(f\\(ea\\)?t\\\\.[^\\])*\\)))  
$set(artist,$rreplace(%artist%,\\s+\\\\(?f\\(ea\\)?t\\\\.[^\\])*\\\\)?,))  
$set(title,$if(%_feat_title%,%title% \\(%_feat_title%\\),%title%))
```

On peut faire de même pour déplacer les artistes vedettes de l'artiste de l'album au titre de l'album:

```
$set(_feat_album,$rsearch(%albumartist%,\\s+\\\\(?\\(f\\(ea\\)?t\\\\.[^\\])*\\)))  
$set(albumartist,$rreplace(%albumartist%,\\s+\\\\(?f\\(ea\\)?t\\\\.[^\\])*\\\\)?,))  
$set(album,$if(%_feat_album%,%album% \\(%_feat_album%\\),%album%))
```

Conserver le nom de fichier original

La balise `originalfilename` est censée contenir le nom de fichier que le fichier avait à l'origine. Par défaut, Picard ne définit ni ne modifie cette balise. Si vous voulez sauvegarder cette information, le script suivant peut être utilisé:

```
$set(originalfilename,$if2(%originalfilename%,%_filename%.%_extension%))
```

Ceci gardera toute balise existante `originalfilename`. Mais si cette balise n'est pas encore présente, elle sera remplacée par le nom du fichier courant. Comme cela se produit avant que le fichier ne soit enregistré, le nom original du fichier avant que Picard ne le modifie peut être préservé.

13.3 Ordre de traitement

Afin d'utiliser efficacement les plugins et les scripts, il est important de comprendre quand chacun est traité par rapport aux autres. En règle générale, les plugins sont toujours traités avant les scripts. Les plugins du même type seront exécutés dans l'ordre en fonction de la priorité spécifiée lors de l'enregistrement du plugin.

13.3.1 Commencez

Au démarrage du programme, les plugins avec les hooks suivants sont traités et toutes les fonctionnalités supplémentaires qu'ils fournissent seront disponibles immédiatement:

- Formats de fichier
- Fournisseurs d'art de couverture
- Fonctions de script de tagger
- Actions du menu contextuel
- Pages d'options

13.3.2 Chargement d'une version

Lorsque les données sont chargées à partir de MusicBrainz (alors que l'album affiche l'état de « chargement » dans le volet de droite), les éléments suivants sont traités:

- Plugins du processeur de métadonnées
- Balisage des scripts

Les plugins ont accès aux données brutes chargées depuis MusicBrainz et sont traités avant les scripts, dans l'ordre de priorité défini lors de l'enregistrement du plugin.

Les scripts sont traités dans l'ordre défini dans le menu Options.

Note: Les scripts de balisage sont toujours exécutés sur les métadonnées chargées depuis MusicBrainz, et exactement après le chargement des données et avant que les fichiers ne soient mis en correspondance. Ils sont l'une des dernières étapes du processus de chargement. Les scripts de balisage n'ont pas accès aux métadonnées stockées dans les fichiers existants.

13.3.3 Chargement de fichiers musicaux

Après le chargement d'un fichier dans Picard, les plugins enregistrés avec `file_post_load_processor()` sont exécutés. Cela pourrait, par exemple, être utilisé pour charger des données supplémentaires pour un fichier.

13.3.4 Ajout / suppression de fichiers

Après qu'un fichier a été ajouté à une piste (sur le volet de droite de Picard), les plugins enregistrés avec `file_post_addition_to_track_processor()` sont exécutés.

Après qu'un fichier a été supprimé d'une piste (sur le volet droit de Picard), les plugins enregistrés avec `file_post_removal_from_track_processor()` sont exécutés.

13.3.5 Sauvegarde des fichiers

Lorsque les fichiers sont enregistrés, pour chaque fichier, le script de dénomination de fichier est d'abord exécuté pour déterminer le chemin de destination et le nom du fichier. Notez que ce script n'a aucun effet sur les valeurs de balises écrites dans le fichier de sortie.

Une fois qu'un fichier a été enregistré, les plugins enregistrés avec `file_post_save_processor()` sont exécutés. Cela peut, par exemple, être utilisé pour exécuter un post-traitement supplémentaire sur le fichier ou écrire des données supplémentaires. Notez que les métadonnées du fichier sont déjà les métadonnées nouvellement enregistrées.

13.3.6 Suppression d'albums

Après qu'un album a été supprimé de Picard, les plugins enregistrés avec `album_post_removal_processor()` sont exécutés.

13.3.7 Menus contextuels

Les scripts de balisage individuels peuvent être exécutés à la demande à partir du menu contextuel affiché lors d'un clic droit sur un fichier, un album, une piste, un cluster ou une liste de cluster.

Dépannage

14.1 Dépannage général

14.1.1 Obtenir de l'aide

Si vous rencontrez des problèmes lors de l'utilisation de Picard, veuillez d'abord consulter les ressources suivantes:

- Pour des informations générales sur l'utilisation, consultez la documentation [Utiliser Picard](#) et le [guide de démarrage rapide illustré](#).
- Lisez la [section FAQ](#) pour les questions et problèmes courants.
- Consultez les [forums de la communauté](#).
- Consultez la [page de téléchargement](#) pour une version plus récente de Picard qui pourrait résoudre votre problème.
- Si le problème est lié à un plugin, vérifiez les [Plugins Picard](#) pour les versions mises à jour du plugin.

14.1.2 Signaler un bogue

Si vous pensez avoir trouvé un bogue, veuillez vérifier si vous utilisez la dernière version de Picard et si le bogue a déjà été signalé dans le [bug tracker](#). Si vous n'êtes pas sûr ou ne souhaitez pas consulter les tickets existants, demandez d'abord sur les forums de la communauté.

Si vous êtes toujours convaincu d'avoir trouvé un nouveau bogue, ouvrez un [nouveau ticket](#) fournissant les informations suivantes:

- Quelle version de Picard utilisez-vous? (« Affecte la version » dans le formulaire)
- Quel système d'exploitation utilisez-vous? (« Environnement » dans le formulaire)
- Qu'avez-vous fait lorsque le bogue est survenu?
- Que s'est-il réellement passé et à quoi vous attendiez-vous?
- Si vous utilisez des plugins, quels plugins avez-vous activés?

14.1.3 Récupération des journaux

Pour de nombreux bogues, cela aide les développeurs à avoir un journal de Picard. Vous pouvez voir le journal en allant dans « *Aide → Afficher le journal* ». Vous pouvez également obtenir un journal de débogage complet (mieux car il contient des informations plus détaillées) en démarrant Picard avec `-d` comme argument de ligne de commande. Si vous utilisez Windows, vous pouvez modifier la cible de votre raccourci (*raccourci clic droit → Propriétés*) en:

```
"C:\Program Files\MusicBrainz Picard\picard.exe" -d
```

Le fait de coller ce journal dans votre message de forum ou votre ticket de bogue peut aider les développeurs et les autres utilisateurs à résoudre votre problème plus rapidement.

Avertissement: N'oubliez pas de supprimer d'abord toutes les informations personnelles et confidentielles telles que l'identifiant d'utilisateur, les mots de passe ou les jetons d'autorisation avant de publier ou de soumettre une sortie de journal.

14.2 Picard ne démarre pas

Si vous constatez que Picard ne démarre pas, il y a quelques raisons possibles courantes et des choses à essayer pour corriger le problème. Avant de faire quelque chose de radical, il est recommandé d'essayer de démarrer Picard à partir de la ligne de commande avec l'option `-d` pour générer la journalisation du débogage. Ce processus est décrit dans la section [Dépannage général](#). Si les journaux résultants ne fournissent aucun indice sur le problème, il peut s'agir de l'un des éléments suivants:

Les fichiers du programme sont corrompus

Si vous pensez que cela peut être le problème, la première chose (et la plus simple) à essayer est de réinstaller le programme. Cela devrait résoudre tous les problèmes potentiels de corruption de fichiers. Si Picard ne démarre toujours pas, il est peu probable que ce soit le problème.

Un fichier de plugin est corrompu ou est incompatible

Pour vérifier si l'un des fichiers du plugin est corrompu ou, dans le cas d'une mise à jour récente vers un plugin ou Picard, un plugin n'est pas compatible, vous devriez essayer de supprimer tous les plugins puis démarrer Picard. Étant donné que vous ne pourrez pas désactiver ou supprimer les plugins en utilisant les paramètres "Option" de Picard, vous devrez les supprimer manuellement. Les plugins peuvent être situés dans un sous-répertoire `plugins` du répertoire où est stocké le fichier programme Picard, ou dans un répertoire spécifique à l'utilisateur:

- *Windows*: `C:\Users\user\AppData\Local\MusicBrainz\Picard\plugins`
- *macOS*: `~/Library/Preferences/MusicBrainz/Picard/plugins`
- *Linux*: `~/.config/MusicBrainz/Picard/plugins`

Une fois que vous avez localisé les fichiers du plugin, ils doivent être supprimés du répertoire `plugins` et déplacés vers un répertoire temporaire. Alors essayez de démarrer Picard. Si le programme démarre, vous devriez essayer de restaurer les fichiers du plugin à partir de votre répertoire temporaire un par un, et vérifier si Picard va démarrer. Cela aidera à identifier le plugin à l'origine du problème.

Le fichier de paramètres d'options est corrompu ou est incompatible

Pour vérifier si le fichier de paramètres d'options de Picard est corrompu ou, dans le cas d'une mise à niveau récente vers Picard, il n'est pas compatible, vous devez essayer de supprimer le fichier de paramètres, puis démarrer Picard. Si Picard est démarré sans trouver son fichier de paramètres de configuration, il en créera un nouveau en utilisant les paramètres par défaut. Le fichier de paramètres s'appelle `Picard.ini` et se trouve dans un répertoire spécifique à l'utilisateur:

- *Windows*: `C:\Users\user\AppData\Roaming\MusicBrainz`
- *macOS*: `~/Library/Preferences/MusicBrainz`
- *Linux*: `~/.config/MusicBrainz`

Encore une fois, il est recommandé de déplacer le fichier vers un répertoire temporaire afin qu'il puisse être récupéré si cela ne s'avère pas être la cause du problème.

Il y a vraiment un bug en Picard

Si ce problème a commencé juste après la mise à jour de Picard, malgré tous les tests effectués avant la sortie d'une nouvelle version, il se peut qu'il s'agisse bien d'un bogue. Dans ce cas, vous devez d'abord essayer de réinstaller la version précédente pour vous assurer qu'elle fonctionne et que le problème ne se produit qu'avec la nouvelle version. Ensuite, vous devez signaler le problème, en suivant les étapes décrites dans la rubrique « Signaler un bogue » de la section [Dépannage général](#). Veillez à inclure autant d'informations que possible, ce qui aidera les développeurs à localiser et à résoudre le problème.

14.3 Il n'y a pas de pochette

Il y a deux problèmes différents qui relèvent souvent de ce sujet:

14.3.1 Picard ne trouve et ne télécharge aucune pochette

Aucun fournisseur de pochettes n'a été activé dans les paramètres de configuration

Confirmez que les paramètres « *Options* → *Options...* → *Cover Art* » ont au moins un fournisseur de jaquettes activé. Veuillez consulter la section [*Fournisseurs d'art de couverture*](#) pour plus d'informations.

Aucune pochette n'est disponible auprès des fournisseurs sélectionnés

Il est possible que la version sélectionnée ne dispose d'aucune pochette disponible auprès des fournisseurs de pochettes activés. Si une image de couverture est affichée pour la version sur le site Web MusicBrainz, il est possible que l'image du groupe de versions soit affichée ou qu'elle soit fournie par le biais d'un accord de fournisseur tiers. Parfois, cela peut être résolu en activant les options de fournisseur « Groupe de versions CAA » et « URL de couvertures autorisées ».

Le fournisseur sélectionné n'est pas disponible actuellement

Parfois, le serveur fournissant la pochette (par exemple: archive.org) n'est pas disponible ou les serveurs miroir n'ont pas encore été synchronisés avec les dernières mises à jour. Dans ce cas, vous devrez peut-être attendre quelques minutes avant de réessayer votre demande. L'examen des détails dans le journal de Picard permet souvent de déterminer si c'est le problème ou non.

La pochette est toujours en attente de modification

Si la pochette a été ajoutée récemment, la modification ajoutant l'image n'a peut-être pas encore été acceptée et appliquée. Vous pouvez demander à Picard d'utiliser la pochette des modifications en attente en désactivant l'option « Télécharger uniquement les images approuvées » dans la sous-section Archives de la pochette des paramètres « *Options* → *Options...* → *Art de couverture* ». Veuillez consulter la section [*Cover Art Archive*](#) pour plus d'informations.

14.3.2 La couverture enregistrée avec les fichiers n'est pas affichée

Le lecteur ne prend pas en charge la pochette intégrée

Vérifiez que votre lecteur prend en charge les images de couverture intégrées. Cette prise en charge n'est pas universelle pour tous les lecteurs. Certains lecteurs prennent en charge les images intégrées, d'autres les images stockées sous forme de fichiers dans le répertoire (par exemple: `cover.jpg` ou `folder.jpg`), et d'autres encore les deux. Picard vous permet de spécifier comment les images de couverture doivent être enregistrées. Veuillez consulter la section *Emplacement* des options Coverart pour plus de détails.

Vous devez également confirmer que votre lecteur prend en charge la version des balises en cours d'écriture.

Voir aussi:

Pour plus d'informations, veuillez consulter: [Fichiers AAC](#) / [Fichiers AC3](#) / [Fichiers ID3](#) / [Fichiers WAVE](#)

Image de couverture intégrée trop grande

Même si votre image de couverture a été correctement intégrée dans le fichier, il est possible que votre lecteur rencontre des difficultés avec les images intégrées dépassant une certaine taille. Si tout le reste échoue, vous pouvez essayer d'utiliser une image avec une taille de fichier plus petite.

14.4 Les balises ne sont pas mises à jour ou enregistrées

Il existe généralement quatre raisons pour lesquelles les balises peuvent ne pas être écrites ou mises à jour lorsque les fichiers sont enregistrés:

L'enregistrement des balises n'a pas été activé dans les paramètres de configuration

Vérifiez que le paramètre « *Options → Enregistrer les variables* » a été activé. Voir [Options d'action](#) pour plus d'informations.

Les balises sont définies dans le script de dénomination de fichier

Les balises créées ou mises à jour dans le script de dénomination de fichier ne seront pas écrites dans les fichiers de sortie. Ce script est uniquement utilisé pour développer le nom de fichier et la structure de répertoires pour la sortie. Si vous souhaitez définir ou mettre à jour une valeur de balise dans un script, elle doit se trouver dans un script de balisage. Veuillez consulter la section [Scripts](#) pour plus d'informations sur les différents types de scripts.

Les balises commencent par un trait de soulignement

Les balises dont les noms commencent par un trait de soulignement, quelle que soit la façon dont elles sont créées, ne seront pas écrites dans les fichiers

de sortie. Celles-ci sont considérées comme des variables à utiliser dans Picard plutôt que des balises. Veuillez consulter la section [Balises & Variables](#) pour plus d'informations concernant la différence entre les balises et les variables.

Le type de fichier ne prend pas en charge l'écriture de balises

Confirmez que le type de fichier que vous écrivez prend réellement en charge les balises à écrire. Tous les types de fichiers ne prennent pas en charge toutes les balises prises en charge par Picard.

Veuillez consulter la section [Annexe B: Mappage de balises](#) pour plus de détails sur les balises prises en charge par différents formats de fichiers.

14.5 Les fichiers ne sont pas enregistrés

Il existe deux scénarios typiques dans lesquels les fichiers ne sont pas enregistrés:

Après avoir sélectionné les fichiers dans le volet de droite, vous voyez un arrêt rouge comme une icône

Cela indique qu'une erreur s'est produite lors de l'enregistrement. Dans la plupart des cas, les gens le voient, c'est parce que les fichiers qu'ils veulent enregistrer sont protégés en écriture (soit ont l'indicateur de lecture seule, soit ont de mauvaises autorisations). Vérifiez que les fichiers ne sont pas protégés en écriture et que vous disposez des autorisations appropriées avant de réessayer.

Les problèmes d'autorisation semblent être plus courants lorsque Picard a été installé à l'aide de Flatpak, ou lorsque les fichiers sont lus ou écrits sur un partage samba sur le réseau.

Une autre possibilité est que la longueur totale du chemin de destination et du nom de fichier dépasse la longueur maximale autorisée par le système d'exploitation. Si vous avez un chemin et un nom de fichier extrêmement longs, essayez de le raccourcir pour le voir, cela permet au fichier d'être enregistré.

Dans le volet de droite, vous ne voyez qu'une icône de note de musique devant les pistes

Cela signifie qu'il ne s'agit que des données de piste de MusicBrainz, mais qu'aucun fichier n'y est associé. Dans ce cas, le bouton Enregistrer est désactivé. Assurez-vous que les fichiers correspondent correctement aux pistes avant de réessayer de sauvegarder. Veuillez consulter les sections [Mise en correspondance des fichiers aux pistes](#) et [Enregistrement des fichiers mis à jour](#) pour plus d'informations.

Une troisième possibilité, bien que très rare, est que vous essayez de définir une balise avec une clé invalide. Si les deux solutions ci-dessus ne résolvent pas votre problème,

essayez de revoir toutes les balises à écrire pour voir s'il y en a qui ne semblent pas valides.

14.6 Picard vient de cesser de travailler

Il y a généralement deux raisons pour lesquelles Picard fonctionnera très lentement ou semblera bloqué:

Traitement d'un grand nombre de fichiers à la fois

Lors du traitement d'un grand nombre de fichiers en un seul lot, Picard peut rencontrer des problèmes soit en raison du traitement de chaque fichier (par exemple: empreinte digitale AcoustID) soit pendant les recherches après le clustering ou l'empreinte digitale en raison de toutes les demandes d'informations à l'API du serveur MusicBrainz. comme téléchargement de la pochette. Même si Picard continue de se frayer un chemin à travers le backlog, l'interface utilisateur peut devenir non réactive et sembler que le programme est bloqué ou bloqué.

L'impact du traitement de fichiers par lots importants est exacerbé lors de l'utilisation de plugins qui effectuent des appels de demande d'informations supplémentaires à l'API du serveur MusicBrainz.

Si vous traitez une grande bibliothèque de fichiers, il est généralement plus efficace de traiter des lots plus petits (par exemple: 200 fichiers) à la fois, en récupérant d'abord les informations à l'aide d'un cluster et d'un processus de recherche, puis en traitant les fichiers restants sans correspondance à l'aide de l'analyse processus. Veuillez consulter la section *Récupération des informations sur l'album* pour plus d'informations.

Traitement des fichiers via une connexion réseau

Si vous traitez des fichiers via une connexion réseau, cela peut avoir un impact sur la vitesse à laquelle Picard fonctionne en raison de la différence de vitesse entre une connexion réseau et un lecteur local. Dans ce cas, le débit peut être amélioré en copiant d'abord les fichiers source sur un lecteur local, en procédant avec Picard, puis en déplaçant les fichiers résultants vers le lecteur réseau.

14.7 macOS montre que l'application est endommagée

Sur macOS 10.12 et 10.13, il y a eu des rapports indiquant que parfois l'application MusicBrainz Picard ne peut pas être démarrée et macOS affiche un message d'erreur:

« MusicBrainz Picard.app » est endommagé et ne peut pas être ouvert. Vous devez le déplacer vers la corbeille.

Cela semble se produire principalement après le déplacement du fichier dans le dossier Applications et semble être causé par Gatekeeper marquant par erreur l'application comme endommagée. Pour résoudre le problème, ouvrez un terminal et exécutez:

```
xattr -c "/Applications/MusicBrainz Picard.app"
```

Cela effacera l'application marquée comme endommagée. Si vous avez placé l'application dans un emplacement différent, alors /Applications ajustez le chemin dans la commande ci-dessus en conséquence.

CHAPTER 15

Questions fréquemment posées

Certaines des questions les plus souvent posées ont été traitées dans les sections suivantes. Ceux-ci ont été organisés en groupes en fonction de l'opération en cours.

15.1 Utiliser Picard

15.1.1 Comment marquer des fichiers avec Picard?

Une section distincte explique le processus de marquage. Veuillez consulter [Utiliser Picard](#) pour plus de détails.

15.1.2 Le « Tagger » vert icône a disparu de MusicBrainz.org, comment le récupérer?

Cette icône apparaît lorsqu'une recherche manuelle est effectuée via Picard en utilisant « *Outils → Recherche* ».

Alternativement, le paramètre `?tport=8000` peut être ajouté à la fin de presque toutes les URL MusicBrainz et les icônes de tagger vertes continueront à apparaître à partir de là.

15.1.3 J'essaye de charger une version en Picard, mais tout ce que je vois est « Impossible de charger les erreurs d'album ». Quoi de neuf?

Si vous obtenez « Impossible de charger les erreurs d'album » pour les versions en Picard, cela peut se produire pour plusieurs raisons. Vérifiez les éléments suivants:

1. Le problème persiste-t-il pour une version donnée?

Essayez d'attendre une minute ou deux, voire un peu plus longtemps, puis réessayez avec un clic droit, « Actualiser ». Parfois, les serveurs sont simplement surchargés et rejettent temporairement les demandes.

2. La version a-t-elle été supprimée de MusicBrainz?

Si vous re-taggez des fichiers précédemment balisés avec Picard et que vous obtenez cette erreur, la version a peut-être été supprimé. Essayez de cliquer avec le bouton droit de la souris et d'utiliser l'option « Rechercher dans le navigateur » pour afficher la version sur le site Web. Si vous ne le trouvez pas, il se peut qu'il ait été supprimé. Cela peut être dû au fait que vous avez balisé une version en attente qui a été rejetée ou que vous avez balisé une version qui a été supprimée parce que les éditeurs ont décidé qu'il ne s'agissait pas d'une version valide. Cela peut arriver pour les compilations homebrew, les mauvais rips torrent ou pirate, les versions « avancées » ou les versions très mal ajoutées. Habituellement, il y aura une version alternative sur laquelle vous pouvez étiqueter, que vous pouvez trouver en recherchant ou en effectuant une autre recherche groupée de Picard. Si vous ne trouvez pas de remplaçant et pensez qu'il a été supprimé injustement, [soumettez une nouvelle version](#), en fournissant la preuve de la liste des pistes et autant d'informations que possible prouver qu'il est authentique et il peut être accepté à nouveau.

15.1.4 J'utilise macOS, où sont mes dossiers réseau ou lecteurs?

Ils devraient apparaître dans les boîtes de dialogue d'ajout de fichier et d'ajout de dossier, mais ils ne sont pas visibles par défaut dans le volet du navigateur de fichiers. Si vous voulez les voir dans le volet du navigateur de fichiers, faites un clic droit dans le volet et sélectionnez « Afficher les fichiers cachés ». Ils devraient alors être visibles dans le dossier /Volumes.

15.1.5 macOS montre que l'application est endommagé. Comment puis-je diriger Picard?

Sur macOS 10.12 et 10.13, il y a eu des rapports indiquant que parfois l'application MusicBrainz Picard ne peut pas être démarrée et macOS affiche un message d'erreur:

« MusicBrainz Picard.app » est endommagé et ne peut pas être ouvert. Vous devez le déplacer vers la corbeille.

Cela semble se produire principalement après le déplacement du fichier dans le dossier Applications et semble être causé par Gatekeeper marquant par erreur l'application comme endommagée. Pour résoudre le problème, ouvrez un terminal et exécutez:

```
xattr -c "/Applications/MusicBrainz Picard.app"
```

Cela effacera l'application marquée comme endommagée. Si vous avez placé l'application dans un emplacement différent de /Applications ajustez le chemin dans la commande ci-dessus en conséquence.

15.1.6 Picard est installé sur Linux en tant que Snap, comment puis-je accéder à un support amovible?

Picard installé en tant que Snap s'exécute dans un bac à sable et n'a donc pas un accès complet à tous les fichiers et dossiers de votre système. Par défaut, Picard a accès à votre dossier personnel. Vous pouvez en outre lui donner accès à un support amovible en exécutant la commande suivante sur un terminal:

```
snap connect picard:removable-media
```

15.2 Formats de fichiers

15.2.1 Quels formats Picard prend-il en charge?

Picard prend en charge les formats de fichiers suivants:

- MPEG-1 Audio (.mp3, .mp2, .m2a)
- MPEG-4 Audio (.m4a, .m4b, .m4p, .m4v, .mp4)
- Windows Media Audio (.wma, .wmv, .asf)
- Microsoft WAVE (.wav)
- The True Audio (.tta)
- FLAC (.flac)
- Audio Interchange File Format (.aiff, .aif, .aifc)
- Musepack (.mpc, .mp+)

- WavPack (.wv)
- OptimFROG (.ofr, .ofs)
- Monkey's Audio (.ape)
- Tom's lossless Audio Kompressor (.tak)
- Speex (.spx)
- Generic Ogg files (.ogg)
- Ogg FLAC (.ogg, .ogv)
- Ogg Theora (.ogg, .oga)
- Ogg Opus (.opus)
- Ogg Audio (.oga)
- Ogg Video (.ogv)
- ADTS stream / AAC (.aac)
- AC-3 (.ac3, .eac3)
- Direct Stream Digital (.dff, .dsf)

Note: Les fichiers WAVE n'ont pas de norme pour un marquage approprié. Picard utilise des balises ID3v2 pour baliser les fichiers WAVE, mais cela n'est pas pris en charge par tous les logiciels. Pour assurer la compatibilité avec un logiciel qui ne prend pas en charge les balises ID3v2 dans les fichiers WAVE, des balises RIFF INFO supplémentaires peuvent être écrites dans les fichiers. RIFF INFO n'a qu'une prise en charge limitée des balises et des encodages de caractères.

15.2.2 Quels formats Picard prend-il en charge?

Picard est destiné à prendre en charge à terme tous les formats (y compris les empreintes digitales), mais il s'agit d'un processus complexe (sans doute sans fin), et prendra un certain temps.

15.2.3 Quelles balises Picard peut-il écrire dans mes fichiers?

Voir la section [Balises & Variables](#) pour plus d'informations sur les champs MusicBrainz que Picard écrit dans les balises. [Picard Tag Mapping](#) contient des informations plus techniques sur la manière dont ils sont davantage mappés dans chaque format de tag.

15.2.4 Comment puis-je modifier plusieurs balises à la fois? Pourquoi n'est-ce pas plus facile à faire?

Veuillez comprendre que Picard n'est pas conçu comme un éditeur de balises à usage général. Son objectif principal est de récupérer les données MusicBrainz maintenues par la communauté pour les écrire dans vos balises. Certains objectifs secondaires comprennent:

- autoriser la personnalisation basée sur des règles de ces données à l'aide de scripts et de plugins
- encourager les utilisateurs à créer un compte et à corriger et mettre à jour les données via le site Web MusicBrainz, partageant ainsi leur travail avec le reste de la communauté plutôt que de simplement réparer leurs tags localement.

À cette fin, Picard ne se concentrera probablement jamais autant sur l'édition manuelle en masse des balises que d'autres éditeurs à usage général (par exemple: Mp3tag, foobar2000, ou même de nombreux gestionnaires de bibliothèques tels que iTunes, Windows Media Player et MediaMonkey). Cela ne veut pas dire que l'équipe n'acceptera pas de correctifs dans ce domaine!

Cela dit, il est toujours possible d'éditer plusieurs balises à la fois dans Picard en suivant les étapes:

1. Cliquez et sélectionnez plusieurs fichiers avec Ctrl ou Shift.
2. Faites un clic droit sur l'un d'eux, puis cliquez sur Détails ...
3. Dans la boîte de dialogue contextuelle, vous pouvez voir les balises, avec des entrées qui indiquent où les balises sont différentes d'un fichier à l'autre. Vous pouvez modifier ou ajouter de nouvelles balises ici.
4. En quittant la boîte de dialogue, vous avez modifié les balises en mémoire. Vous devez cliquer sur Enregistrer pour conserver ces modifications dans vos fichiers.

Ce processus devrait fonctionner dans les deux volets.

15.2.5 Le lecteur audio intégré ne peut pas lire mon fichier. Quels formats prend-il en charge?

Les formats pris en charge par le lecteur audio intégré dépendent des formats pris en charge par votre système d'exploitation.

Windows:

Les formats pris en charge dépendent des codecs installés. Selon la version de Windows, certains codecs sont préinstallés, mais vous pouvez installer des codecs supplémentaires.

Vous voudrez peut-être installer les [Directshow Filters for Ogg](#) pour ajouter la prise en charge des fichiers Ogg Vorbis, Ogg Speex, Ogg Theora, Ogg FLAC, FLAC natif et WebM.

Voir aussi:

Des informations supplémentaires sont disponibles dans [FAQ sur les codecs de Microsoft](#).

Linux:

Sur les systèmes Linux, le lecteur utilise GStreamer qui prend en charge les formats audio les plus courants, bien que certaines distributions puissent exclure certains codecs en raison de problèmes de licence. Pour une prise en charge du format le plus large, assurez-vous d'installer tous les plugins GStreamer disponibles pour votre distribution.

15.2.6 J'utilise Fedora. Pourquoi la prise d'empreintes acoustique ne fonctionne-t-elle pas?

La prise d'empreintes acoustiques dans Picard utilise un outil appelé **fpcalc**, qui n'est pas disponible dans Fedora. Vous pouvez l'obtenir en installant le paquetage chromaprint-tools depuis le dépôt [RPM Fusion](#). Cette fonctionnalité n'est pas contenue dans le paquet principal de Fedora picard car elle nécessite le paquet ffmpeg qui [n'est pas distribué par Fedora](#). Après avoir activé le dépôt de RPM Fusion « rpmfusion-free », installez le paquet (en tant que root) en utilisant:

```
yum install chromaprint-tools
```

15.3 Configuration

15.3.1 Où est le Picard configuration enregistré?

Picard enregistre la configuration dans le fichier Picard.ini. Son emplacement dépend du système d'exploitation:

Windows:

%APPDATA%\MusicBrainz\Picard.ini

Ce sera généralement C:\Users\YourUserName\AppData\Roaming\MusicBrainz, où YourUserName doit être remplacé par votre nom d'utilisateur Windows actuel.

macOS, Linux et autres systèmes de type Unix:

\$HOME/.config/MusicBrainz/Picard.ini

15.3.2 J'ai marqué un fichier en Picard, mais iTunes ne voit pas les balises!

Tout d'abord, vous devez forcer iTunes à relire les informations de vos balises et à mettre à jour sa bibliothèque. Ceci est discuté dans le [iTunes Guide](#).

De plus, iTunes a un bogue connu dans son implémentation ID3v2.4, qui le rend incapable de lire ces balises si elles contiennent également un code de couverture intégrée. Pour contourner le problème, vous pouvez configurer Picard pour écrire des balises ID3v2.3.

15.3.3 Mes balises sont truncées à 30 caractères dans Windows Media Player!

Avant la version 0.14, les paramètres par défaut de Picard étaient d'écrire les balises ID3v2.4 et ID3v1 dans les fichiers. WMP ne peut pas lire ID3v2.4, il revient donc à ID3v1 qui a une limitation de 30 caractères par titre. Pour résoudre ce problème sur les versions antérieures à 0.14, configurez Picard pour qu'il écrive des balises ID3v2.3 à la place.

À partir de la version 0.14, les paramètres par défaut ont été modifiés en ID3v2.3 et cela ne devrait plus être un problème.

15.3.4 Comment dire à Picard quel navigateur utiliser?

Sous Windows, macOS, GNOME et KDE, Picard utilise le navigateur par défaut qui a été configuré pour le système. Sur d'autres systèmes, vous pouvez utiliser la variable d'environnement BROWSER.

Par exemple:

```
export BROWSER="firefox '%s' &"
```

Une autre approche qui fonctionne dans certains systèmes GNU / Linux est la commande suivante:

```
sudo update-alternatives --config x-www-browser
```

Cela devrait vous présenter une liste des navigateurs existants dans votre système, vous permettant de sélectionner celui à utiliser par défaut.

CHAPTER 16

Tutoriels

16.1 Ecriture d'un script de nommage de fichier

Ecrire un script pour organiser et nommer vos fichiers n'est en fait pas si difficile - ne vous laissez pas intimider par tous les "\$", "%" et les parenthèses. Si vous pouvez écrire un modèle comme « **ARTISTE - (ANNÉE) NOM/PISTE DE L'ALBUM - TITRE DE LA CHANSON** » de la façon dont vous voulez que les fichiers et dossiers soient nommés, vous pouvez facilement traduire cela dans le script approprié.

Notez que l'utilisation d'un "/" dans la chaîne de formatage sépare le répertoire de sortie du nom de fichier. La chaîne de mise en forme peut contenir n'importe quel nombre de caractères "/". Tout ce qui précède le dernier "/" est l'emplacement du répertoire, et tout ce qui se trouve après le dernier "/" devient le nom du fichier. Dans notre exemple, nous n'avons qu'un seul caractère "/", ce qui signifie que nous aurons un niveau de répertoire pour l'album qui contiendra les chansons de cet album.

Voyons d'abord ce dont nous avons besoin. Vous voyez une liste des balises disponibles dans la section *Balises de base*. Nous voulons le nom **ARTISTE**, donc les balises disponibles pour cela pourraient être `albumartist` ou `artist`. Cela devrait être le nom d'un dossier d'album, donc `albumartist` ressemble à ce dont nous avons besoin. Pour obtenir la valeur réelle d'une balise, vous devez mettre son nom entre des signes de pourcentage. Alors commençons:

```
%albumartist%
```

Maintenant, nous voulons le **ANNÉE**. Il n'y a pas de balise année, mais il y a date. Utilisons ceci pour le moment. Si nous voulons ajouter du texte supplémentaire comme le « - », écrivez-le simplement. Nous devons faire attention aux parenthèses, car ce sont des variables spéciales dans les scripts. Nous devons les préfixer avec une barre oblique inverse. Ajoutons tout cela:

```
%albumartist% - \(%date%\)
```

Nous voulons maintenant le **NOM DE L'ALBUM**. C'est simple, utilisez simplement album:

```
%albumartist% - \(%date%\) %album%
```

Cela prend en charge la partie répertoire du changement de nom. La partie suivante est le numéro **PISTE DE L'ALBUM** et **TITRE DE LA CHANSON**. Le numéro de piste est disponible sous forme de tracknumber et le titre de la piste est simplement title. En les ajoutant à notre script, nous obtenons:

```
%albumartist% - \(%date%\) %album%/%tracknumber% - %title%
```

Vous pouvez voir que cela ressemble presque au modèle que nous avons dit que nous voulions au début. Ce n'est pas encore parfait pour plusieurs raisons. Que faire s'il y a 10 pistes ou plus sur l'album et qu'elles ne sont pas triées correctement dans la liste du répertoire? De plus, nous obtenons une date complète au lieu de seulement l'année. Enfin, parfois, si vous marquez des fichiers existants, ils peuvent ne pas avoir le jeu albumartist, juste artiste.

Fixons d'abord le numéro de la piste. Nous pouvons nous en occuper en utilisant la fonction `$num` pour ajouter un zéro non significatif au numéro affiché pour les pistes 1 à 9:

```
%albumartist% - \(%date%\) %album%/$num(%tracknumber%,2) - %title%
```

Maintenant, corigeons le **ARTISTE**. Nous pouvons revenir à l'utilisation de artist si albumartist n'est pas disponible en utilisant:

```
$if2(%albumartist%,%artist%) - \(%date%\) %album%/$num(%tracknumber%,2) -  
→ %title%
```

La fonction `$if2` utilise la première valeur qui n'est pas vide, donc si albumartist est vide, il utilise artist à la place.

Pour la balise date, les dates de MusicBrainz sont toujours au format AAAA-MM-JJ. Nous n'avons besoin que de l'année, donc n'obtenons que les 4 premiers caractères avec la fonction `$left`:

```
$if2(%albumartist%,%artist%) - \($left(%date%,4)\) %album%/$num(%tracknumber%,  
→ 2) - %title%
```

Que se passe-t-il s'il n'y a pas d'informations d'étiquette date? Parfois, MusicBrainz n'a pas la date de sortie d'un album, car elle n'est pas encore connue ou n'a pas été entrée dans la base de données. Ce serait bien d'omettre la date entière avec les parenthèses dans ce cas. Utilisons la fonction `$if` pour vérifier si la date est définie:

```
$if2(%albumartist%,%artist%) - $if(%date%,\($left(%date%,4)\) )%album%/$num(  
→ %tracknumber%,2) - %title%
```

Alternativement, nous pouvons entrer un espace réservé tel que « **0000** » si la date est manquante:

```
$if2(%albumartist%,%artist%) - \($if(%date%,$left(%date%,4),0000)\) %album%/  
↪$num(%tracknumber%,2) - %title%
```

Et là vous l'avez - le script final pour nommer vos fichiers développé à partir du modèle que nous avons utilisé comme point de départ.

Voir aussi:

Pour plus d'informations sur les balises et variables disponibles, veuillez consulter la section *Balises & Variables*. Pour plus d'informations sur les fonctions de script disponibles, veuillez consulter la section *Fonctions de script*.

16.2 Compréhension Empreintes acoustiques et AcoustIDs

L'empreinte digitale est la base de tout le système d'identification des chansons AcoustID. L'empreinte digitale audio capture les caractéristiques de l'enregistrement, mais il peut y avoir de légères différences dans l'empreinte digitale des fichiers du même enregistrement causées par des choses telles qu'un encodage ou un débit binaire différent. Les empreintes digitales, ainsi que les métadonnées de la piste, sont soumises au site Web AcoustID où le serveur AcoustID combine des empreintes digitales suffisamment similaires et leur attribue un seul AcoustID. C'est en fait ce qui fait que le système AcoustID fonctionne vraiment pour l'identification audio. Le même enregistrement peut générer de nombreuses empreintes digitales légèrement différentes, mais l'AcoustID représente ce que le service identifie comme étant le même enregistrement pour toutes les empreintes digitales associées.

Ce que fait Picard est le suivant:

1. Lorsque vous cliquez sur « Scan » sur un fichier, Picard génère l'empreinte audio du fichier, en utilisant l'utilitaire de ligne de commande **fpcalc** fourni par AcoustID.
2. Picard utilise cette empreinte digitale pour rechercher un AcoustID sur le serveur AcoustID. Le serveur AcoustID comparera l'empreinte digitale et essaiera de la faire correspondre à un AcoustID existant. Il existe trois possibilités:
 - Il ne trouve pas d'AcoustID. La recherche a échoué.
 - Le serveur AcoustID trouve un AcoustID existant pour l'empreinte digitale soumise, mais il n'est associé à aucun enregistrement MusicBrainz. La recherche a échoué.
 - Le serveur AcoustID trouve un AcoustID existant pour l'empreinte digitale soumise et il est associé à un enregistrement MusicBrainz. Picard fait correspondre le fichier à l'un des enregistrements MusicBrainz liés à l'AcoustID.

Si aucun AcoustID n'a été trouvé, vous pouvez utiliser le bouton « Soumettre » en Picard pour soumettre les empreintes digitales au serveur AcoustID une fois que vous avez fait correspondre les fichiers aux enregistrements appropriés. S'il n'existe aucun AcoustID pour une empreinte digitale, le serveur générera un nouvel AcoustID (ce qui peut prendre un certain temps). Il reliera également l'AcoustID à l'enregistrement MusicBrainz identifié par les métadonnées soumises. Veuillez consulter la section [Soumission d'empreintes acoustiques](#) pour une procédure détaillée étape par étape.

Vous n'avez pas besoin du logiciel d'empreinte digitale AcoustID pour générer manuellement de nouveaux AcoustID. La différence est que le logiciel d'empreintes digitales est conçu pour être exécuté sur des fichiers déjà marqués, donc s'il ne trouve pas d'AcoustID, il effectuera immédiatement la soumission. Pour Picard, l'AcoustID est avant tout un outil d'identification, et comme les fichiers sont considérés comme non étiquetés à ce stade *d'identification*, vous ne pouvez faire la soumission qu'une fois que les fichiers ont été correctement mis en correspondance avec un

enregistrement MusicBrainz. Vous constaterez également qu'après la soumission, Picard ne récupérera pas automatiquement les AcoustIDs nouvellement générés. En effet, la génération peut prendre un certain temps et la réponse reçue du serveur AcoustID ne contient pas les AcoustID nouvellement générés. Cependant, si vous effectuez une autre analyse des fichiers après la soumission, l'AcoustID devrait être disponible.

Note: Si les fichiers sont mis en correspondance en utilisant « *Scan* » puis « *Générer des empreintes digitales* » est utilisé sur eux, la soumission ne sera pas activée, car ils ont déjà été comparés par empreinte digitale. C'est la même situation que d'utiliser simplement « *Scan* », car une fois les fichiers scannés, l'empreinte digitale / l'ID d'enregistrement résultant est mémorisé comme ayant déjà été soumis.

Aussi si vous avez des fichiers correspondant aux pistes et que vous utilisez « *Générer des empreintes digitales* » et que vous êtes en mesure de soumettre avec succès les empreintes digitales, essayer d'utiliser « *Générer des empreintes digitales* » pour les mêmes fichiers et pistes à nouveau ne re-activera la soumission de ces fichiers. La raison est la même: Picard se souvient des combinaisons d'empreintes digitales / ID d'enregistrement déjà soumises. Cependant, le redémarrage de Picard (ou même simplement la suppression et l'ajout de ces fichiers) puis l'utilisation de « *Générer des empreintes digitales* » permettra à nouveau de soumettre.

Les empreintes digitales sont soumises par lots en fonction de la taille des empreintes digitales, mais souvent jusqu'à 200 ou 250 empreintes digitales peuvent être soumises en un seul lot. Une demande de soumission pour un lot peut échouer pour diverses raisons, telles que des problèmes de réseau ou de serveur. Si une demande échoue, toutes les empreintes digitales de ce lot de soumission sont toujours marquées comme n'ayant pas été soumises et la soumission peut être réessayée.

Vous pouvez également utiliser « *Générer des empreintes digitales* » sur des fichiers non correspondants ou correspondants. Cela ne générera que les empreintes acoustiques sans effectuer de recherche sur le serveur AcoustID. Cela signifie également qu'aucune balise AcoustID ne sera créée. Cependant, vous pouvez soumettre ces empreintes digitales si vous associez les fichiers d'empreintes digitales à une piste.

16.3 Gestion de plusieurs pays de sortie.

Certaines sorties, en particulier les sorties numériques, peuvent avoir une très longue liste de pays de sortie, listant parfois tous les pays du monde sauf quelques-uns où la sortie n'est pas officiellement disponible. Picard propose quelques outils pour gérer cela.

Prenons l'exemple de **Bleach**, de Nirvana (MusicBrainz release adab3feb-1822-4d27-a997-db7d6c9688c0).

Par défaut, Picard écrit une seule balise `releasecountry` dans les fichiers. Avant la v2.3.1, Picard remplissait cette balise avec ce que le serveur MusicBrainz retournait comme pays pour la sortie. S'il y avait plusieurs sorties, ce champ `pays` était simplement rempli avec le premier par ordre alphabétique (Afghanistan dans notre exemple). Picard v2.3.1 a introduit quelques options pour mieux gérer ce problème.

16.3.1 Utilisation des pays de diffusion préférés

Si vous configurez les pays de publication préférés dans « *Options → Metadata → Preferred Releases* ». Picard utilisera le premier pays de la liste des pays de diffusion préférés qui se trouve également dans la liste des événements de diffusion. Ainsi, si vous avez configuré les pays de diffusion préférés comme étant l'Europe, le Canada, l'Allemagne et le Royaume-Uni, dans notre exemple, cela signifie que la balise `releasecountry` sera définie sur le Canada.

16.3.2 Utilisation de scripts pour définir un pays différent

Picard v2.3.1 a également ajouté une nouvelle variable `%_releasecountries%`, qui fournit la liste complète des pays de diffusion d'une version sous la forme d'une variable à plusieurs valeurs. Vous pouvez l'utiliser pour définir différentes valeurs pour la balise `releasecountry`.

Par exemple, le script suivant lui attribue la valeur « [International] » s'il y a 10 pays de diffusion ou plus:

```
$if($gte($lenmulti(%_releasecountries%),10),$set(releasecountry,
↪[International]))
```

Bien entendu, vous pouvez ajuster le nombre de pays et le texte de remplacement à votre convenance. Vous pouvez également choisir d'enregistrer la liste entière au lieu d'un seul pays dans cette balise en utilisant le script:

```
$setmulti(releasecountry,%_releasecountries%)
```

Vous préférez peut-être limiter cette liste aux premières entrées. L'exemple suivant utilise uniquement les 6 premiers pays:

```
$setmulti(releasecountry,$slice(%_releasecountries%,0,6))
```

16.3.3 Qu'est-ce qui manque?

Les pays sont actuellement inscrits dans les balises sous leur code pays ISO 3166-1, avec quelques valeurs spéciales ajoutées pour les pays historiques et des choses comme [Europe] ou [Worldwide]. Ces codes ne sont pas toujours facilement reconnaissables ou évidents, comme « DZ » pour l'Algérie ou « DE » pour l'Allemagne. Vous pouvez bien sûr utiliser des scripts pour les rendre plus lisibles. Par exemple, si vous voulez voir « United Kingdom » au lieu de « GB » dans cette balise, utilisez:

```
$if($eq(%releasecountry%,GB),$set(releasecountry,United Kingdom))
```

Cela peut fonctionner si vous ne traitez que quelques pays dans votre collection, ou si vous voulez simplement gérer certains cas particuliers comme l'utilisation de « Europe » au lieu de « XE », comme dans le script suivant:

```
$if($eq(%releasecountry%,XE),$set(releasecountry,Europe))
$if($eq(%releasecountry%,XU),$set(releasecountry,[Unknown]))
$if($eq(%releasecountry%,XW),$set(releasecountry,[Worldwide]))
$if($eq(%releasecountry%,XG),$set(releasecountry,DDR))
```

Un changement a été soumis à Picard pour ajouter une fonction `$countryname()` afin de convertir facilement le code en un nom lisible ; cependant, cette fonction n'est pas prévue avant la version 2.7 de Picard.

16.4 Écriture d'un Plugin

Vous avez une excellente idée pour étendre Picard avec un plugin mais vous ne savez pas par où commencer. Malheureusement, c'est un problème courant qui empêche beaucoup trop de ces grandes idées de voir le jour. Peut-être que ce tutoriel vous aidera à commencer à transformer votre grande idée en réalité.

Les plugins Picard sont écrits en Python, c'est donc le langage de programmation que vous utiliserez. Veuillez vérifier le fichier [INSTALL.md](#) dans le dépôt Picard sur GitHub pour voir la version minimale requise. Il s'agit de Python 3.6 au moment où ce tutoriel a été écrit. Consultez également la [Plugins API](#) pour obtenir des informations supplémentaires, notamment les paramètres passés à chacun des types de fonctions.

Pour les besoins de ce tutoriel, nous allons développer un plugin simple pour sauvegarder les informations sur les arguments fournis par Picard aux plugins de traitement track et release. Cela démontrera comment on accède à ces informations, et fournira un utilitaire que vous pourrez trouver utile lorsque vous développerez vos propres plugins.

La première chose que nous devons inclure est l'information d'en-tête qui décrit le plugin.

```
PLUGIN_NAME = "Example plugin"
PLUGIN_AUTHOR = "This authors name"
PLUGIN_DESCRIPTION = "This plugin is an example"
PLUGIN_VERSION = '0.1'
PLUGIN_API VERSIONS = ['2.2']
PLUGIN_LICENSE = "GPL-2.0-or-later"
PLUGIN_LICENSE_URL = "https://www.gnu.org/licenses/gpl-2.0.html"
```

Ensuite, nous listons les modules qui seront référencés dans notre code. Dans ce cas, nous utiliserons le module `os` pour construire le chemin du fichier de sortie, et le module `json` pour formater le texte du dictionnaire d'arguments pour une meilleure lisibilité. Nous enregistrerons notre fichier de sortie dans le répertoire de base utilisé pour le nommage des fichiers. Nous importons donc le module `config` de Picard, ainsi que le module `log` pour pouvoir écrire des messages de débogage ou d'erreur dans le journal de Picard. Enfin, nous importons les hooks de traitement appropriés et les paramètres de priorité des plugins.

```
import json
import os

from picard import config, log
from picard.metadata import (register_album_metadata_processor,
                             register_track_metadata_processor)
from picard.plugin import PluginPriority
```

Avertissement: Pour assurer une compatibilité maximale, vous ne devez utiliser que des modules Python standard, ou des modules tiers qui sont déjà inclus dans

Picard. Si vous utilisez d'autres modules, alors le plugin ne fonctionnera pas correctement s'il est utilisé sur un système qui n'a pas la bonne version du module installé ou si quelqu'un utilise une version exécutable de Picard.

Maintenant nous pouvons commencer à ajouter le code que nous voulons que Picard exécute. Tout d'abord, nous allons identifier le fichier de sortie pour stocker les informations sur les paramètres fournis par Picard. Il s'agit d'un fichier nommé `data_dump.txt` qui sera stocké dans le répertoire de sortie du nommage des fichiers. Nous trouvons le nom du paramètre de configuration dont nous avons besoin, `move_files_to`, en examinant le code source de Picard pour l'écran de paramétrage des options correspondant. Dans ce cas, il s'agit d'une `TextOption` dans la classe `RenamingOptionsPage` trouvée dans le fichier `picard/ui/options/renaming.py`.

```
file_to_write = os.path.join(config.setting["move_files_to"], "data_dump.txt")
```

La partie suivante est une fonction permettant d'écrire un objet Python dans notre fichier de sortie. Pour permettre à la même fonction d'être utilisée dans différentes situations, nous incluons des paramètres pour identifier le type de ligne (type d'entrée), l'objet à écrire, et des options pour écrire au format JSON et ajouter ou écraser un fichier de sortie existant. Dans notre cas, nous voulons écraser le fichier à chaque fois qu'une nouvelle version est traitée, mais toujours ajouter les informations sur les pistes au fichier.

Nous avons également inclus un contrôle d'erreur pour écrire une entrée dans le journal Picard en cas d'exception.

```
def write_line(line_type, object_to_write, dump_json=False, append=True):
    file_mode = 'a' if append else 'w'
    try:
        with open(file_to_write, file_mode, encoding="UTF-8") as f:
            if dump_json:
                f.write('{0} JSON dump follows:\n'.format(line_type))
                f.write('{0}\n\n'.format(json.dumps(object_to_write,
                                                indent=4)))
            else:
                f.write("{0:s}: {1:s}\n".format(line_type, str(object_to_
write)))
    except Exception as ex:
        log.error("{0}: Error: {1}".format(PLUGIN_NAME, ex))
```

Maintenant, nous incluons les fonctions qui seront appelées lorsque les releases et les tracks seront récupérés par Picard. Le hook de la fonction `release` fournit trois arguments, et le hook de la fonction `track` fournit quatre arguments. Les types d'arguments sont décrits dans la section [Plugins API](#). Le premier argument, `album`, est un objet qui contient des informations sur l'album sélectionné. Voir la classe `Album` dans le fichier `picard/album.py` du code source de Picard pour plus d'informations.

Le second argument, `metadata`, est un objet qui contient les tags et les variables que Picard a assignés pour la version et la piste en cours. C'est ici que vous pouvez ajouter

ou modifier les tags et les variables que Picard met à disposition de l'utilisateur pour les scripts. Voir la classe `Metadata` dans le fichier `picard/metadata.py` du code source de Picard pour plus d'informations.

Les arguments `track` et `release` sont des dictionnaires Python contenant les informations fournies en réponse aux appels de Picard à l'API MusicBrainz. Les informations peuvent différer selon les paramètres *Options de métadonnées* de l'utilisateur pour des choses comme « *Utiliser les relations de validation* » ou « *Utiliser les relations de suivi* ».

```
def dump_release_info(album, metadata, release):
    write_line('Release Argument 1 (album)', album, append=False)
    write_line('Release Argument 3 (release)', release, dump_json=True)

def dump_track_info(album, metadata, track, release):
    write_line('Track Argument 1 (album)', album)
    write_line('Track Argument 3 (track)', track, dump_json=True)
    # write_line('Track Argument 4 (release)', release, dump_json=True)
```

Enfin, nous devons enregistrer nos fonctions afin qu'elles soient traitées avec les événements appropriés. Dans notre cas, nous mettons la priorité à « `HIGH` » pour que nous puissions sortir les informations des paramètres dès qu'elles sont reçues par Picard avant que d'autres plugins aient l'opportunité de les modifier.

```
# Register the plugin to run at a HIGH priority so that other plugins will
# not have an opportunity to modify the contents of the metadata provided.
register_album_metadata_processor(dump_release_info, priority=PluginPriority.
    ↪HIGH)
register_track_metadata_processor(dump_track_info, priority=PluginPriority.
    ↪HIGH)
```

Le fichier de code complet du plugin ressemble à quelque chose comme:

```
PLUGIN_NAME = "Example plugin"
PLUGIN_AUTHOR = "This authors name"
PLUGIN_DESCRIPTION = "This plugin is an example"
PLUGIN_VERSION = '0.1'
PLUGIN_API VERSIONS = ['2.2']
PLUGIN_LICENSE = "GPL-2.0-or-later"
PLUGIN_LICENSE_URL = "https://www.gnu.org/licenses/gpl-2.0.html"

import json
import os

from picard import config, log
from picard.metadata import (register_album_metadata_processor,
                             register_track_metadata_processor)
from picard.plugin import PluginPriority

file_to_write = os.path.join(config.setting["move_files_to"], "data_dump.txt")

def write_line(line_type, object_to_write, dump_json=False, append=True):
```

(suite sur la page suivante)

(suite de la page précédente)

```
file_mode = 'a' if append else 'w'
try:
    with open(file_to_write, file_mode, encoding="UTF-8") as f:
        if dump_json:
            f.write('{0} JSON dump follows:\n'.format(line_type))
            f.write('{0}\n\n'.format(json.dumps(object_to_write, indent=4)))
        else:
            f.write("{0:s}: {1:s}\n".format(line_type, str(object_to_write)))
except Exception as ex:
    log.error("{0}: Error: {1}".format(PLUGIN_NAME, ex))

def dump_release_info(album, metadata, release):
    write_line('Release Argument 1 (album)', album, append=False)
    write_line('Release Argument 3 (release)', release, dump_json=True)

def dump_track_info(album, metadata, track, release):
    write_line('Track Argument 1 (album)', album)
    write_line('Track Argument 3 (track)', track, dump_json=True)
    # write_line('Track Argument 4 (release)', release, dump_json=True)

# Register the plugin to run at a HIGH priority so that other plugins will
# not have an opportunity to modify the contents of the metadata provided.
register_album_metadata_processor(dump_release_info, priority=PluginPriority.
    HIGH)
register_track_metadata_processor(dump_track_info, priority=PluginPriority.
    HIGH)
```

C'est tout pour le code de notre plugin. Maintenant nous devons l'empaqueter afin de pouvoir l'installer dans Picard. Si nous voulons juste l'utiliser localement pour nous-mêmes, le plus simple est de nommer le fichier comme `my_plugin.py`. S'il y a plusieurs fichiers, comme des plugins qui incluent des écrans de réglages supplémentaires, alors les fichiers doivent être enregistrés dans un répertoire tel que `my_plugin` avec le fichier principal nommé `__init__.py`. Le répertoire est ensuite archivé dans un fichier `my_plugin.zip`, dont le nom de fichier est identique au nom du répertoire inclus. Le contenu de l'archive s'affiche comme suit:

```
my_plugin/__init__.py
my_plugin/another_file.py
my_plugin/etc
```

Si vous êtes arrivé jusqu'ici, félicitations ! Vous venez de créer votre premier plugin Picard. Vous avez maintenant un point de départ pour transformer cette grande idée en réalité.

Voir aussi:

Des portions pertinentes du code source de Picard, y compris:

- Modules de paramétrage des options dans `picard/ui/options/` pour les noms utilisés pour accéder aux paramètres.

- La classe `Album` dans le fichier `picard/album.py`.
- La classe `Metadata` et les fonctions d'enregistrement du plugin de traitement des métadonnées dans le fichier `picard/metadata.py`.
- La classe `PluginPriority` dans le fichier `picard/plugin.py`.

16.5 Chargement des versions avec MusicBrainz pour Android

Si vous disposez d'un téléphone Android, vous pouvez utiliser l'application MusicBrainz for Android pour rechercher les sorties par recherche textuelle ou par code-barres et les charger dans Picard sur votre ordinateur.

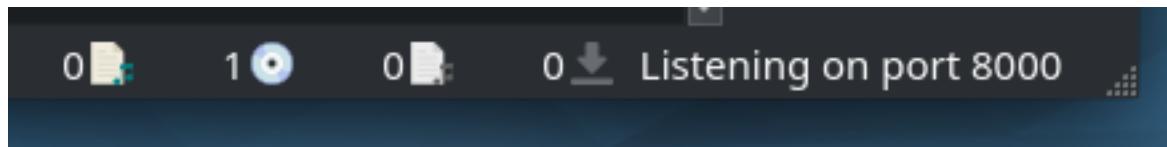
C'est utile par exemple si vous avez déjà extrait vos CD et que vous voulez maintenant étiqueter vos fichiers extraits avec Picard en utilisant exactement les données des disques que vous possédez. Vous pouvez alors utiliser votre téléphone pour scanner les codes-barres de vos CD et charger leurs données dans Picard, puis utiliser ces données pour étiqueter vos fichiers locaux.

Pour que cela fonctionne, votre téléphone et votre ordinateur doivent être connectés au même réseau.

16.5.1 Configuration du Picard

Dans *Options* → *Options...* → *Avancées* → *Réseau*, activez « Intégration du navigateur » et désactivez « Écoute uniquement sur localhost ». Il est recommandé de conserver le port d'écoute sur la valeur par défaut 8000, mais vous pouvez également le modifier.

Une fois les options enregistrées, vérifiez si Picard affiche le message « Listening on port 8000 » dans la barre d'état en bas à droite de la fenêtre principale.



Le numéro de port réel peut varier, mais le numéro par défaut est 8000. Notez le numéro de port, vous en aurez besoin pour configurer l'application Android à l'étape suivante.

Voir aussi:

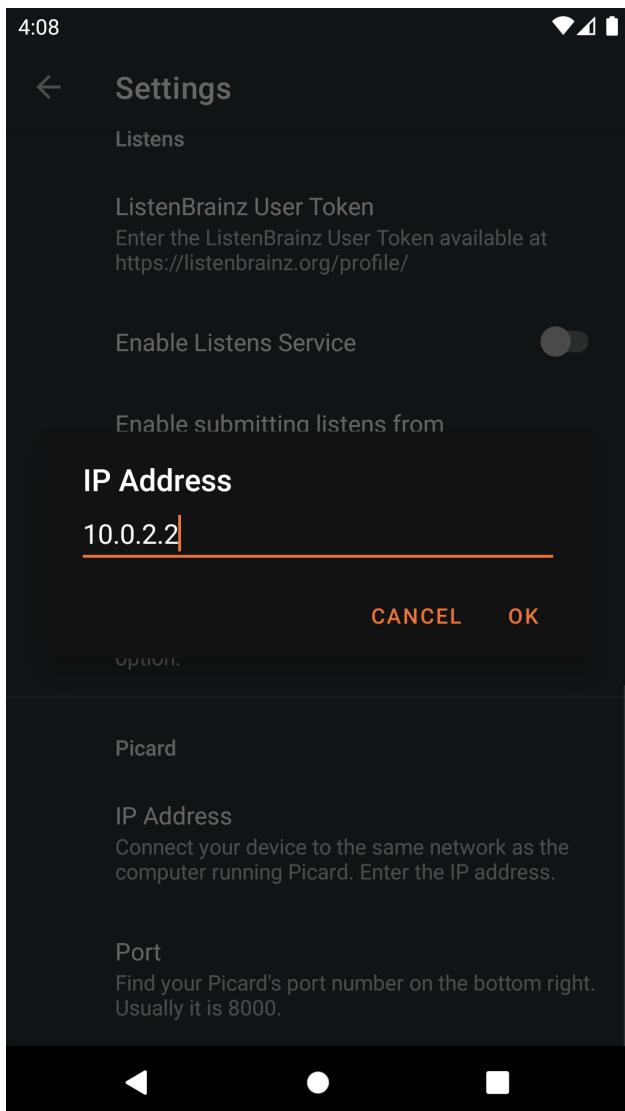
Options de réseau

16.5.2 Installation et configuration de l'application MusicBrainz Android

Installez [MusicBrainz pour Android](#) sur votre téléphone. Vous pouvez télécharger la dernière version de l'application soit sur le [Google Play Store](#) ou [F-Droid](#).

Une fois installée, lancez l'application et appuyez sur l'icône des paramètres en haut à droite. Faites défiler la liste jusqu'aux paramètres de Picard. Pour l'adresse IP, entrez l'adresse IP de votre ordinateur sur lequel Picard fonctionne. Selon la configuration de

votre réseau local, vous pouvez également entrer le nom d'hôte de votre ordinateur au lieu de l'adresse IP.



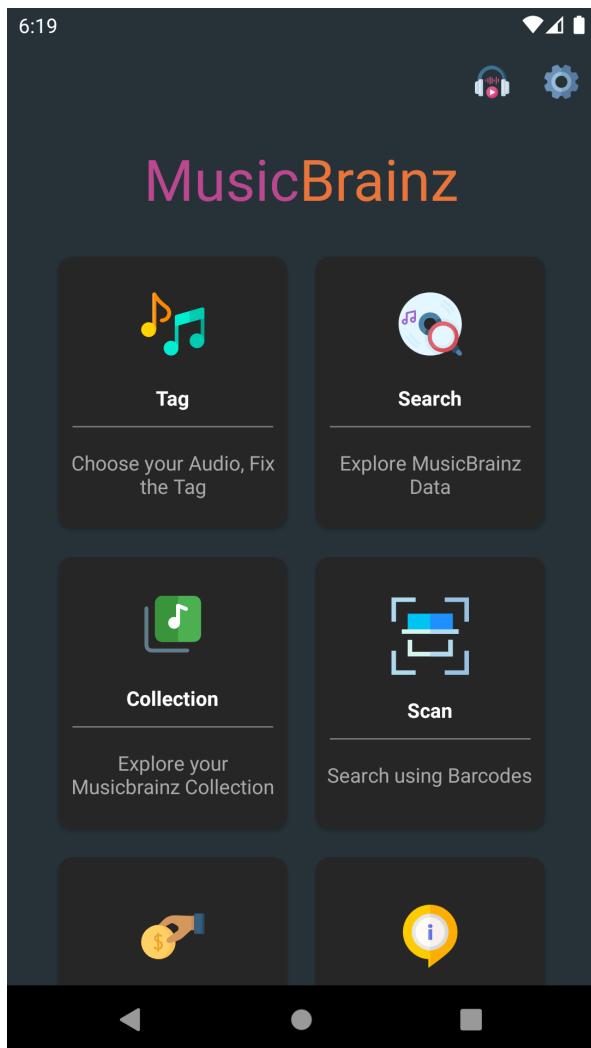
Pour le Port, entrez le port d'écoute de Picard tel qu'il est affiché dans l'écran principal de Picard (voir la section précédente). La valeur par défaut est 8000.

Avant de continuer, assurez-vous que Picard est en cours d'exécution et que le message d'état « Listening on port... » est affiché. Assurez-vous également que votre téléphone est connecté à votre réseau local.

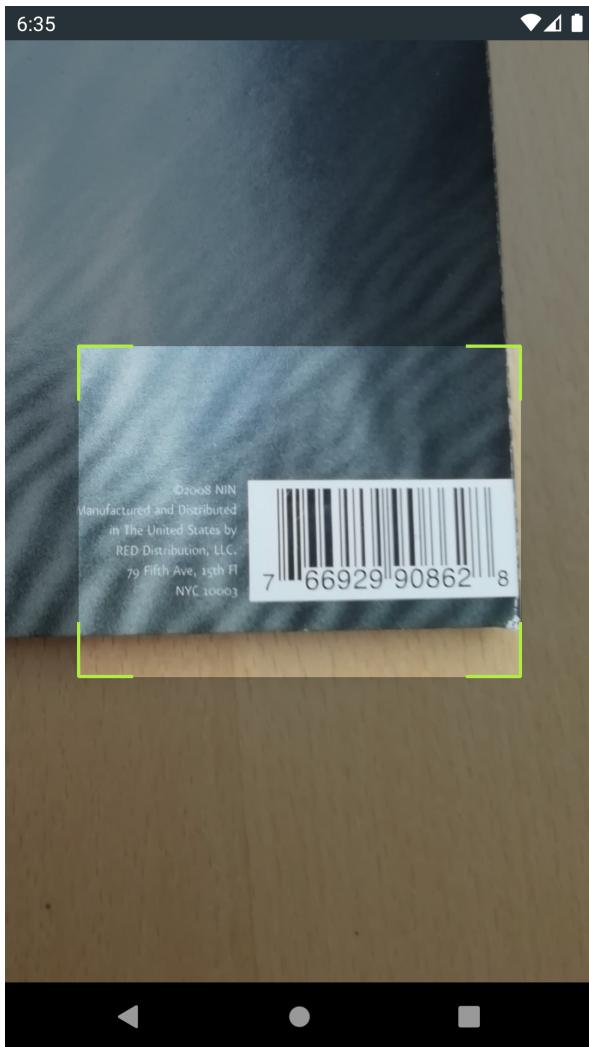
16.5.3 Chargement des communiqués par code-barres

Vous pouvez utiliser votre téléphone comme un scanner de codes-barres pour charger les métadonnées de vos supports physiques:

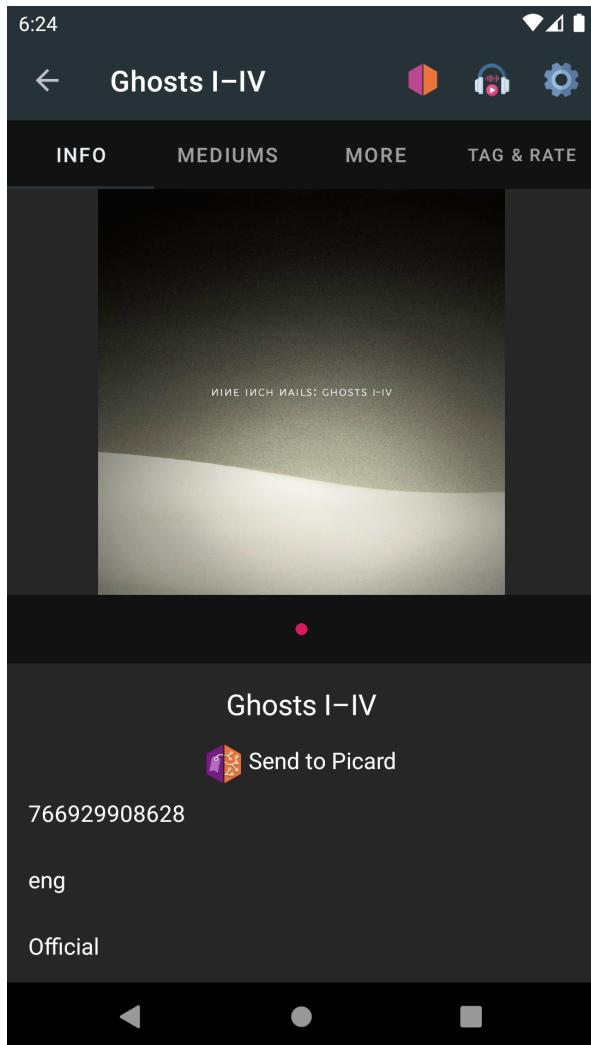
1. Sur l'écran principal de l'application Android, tapez sur « Scan ».



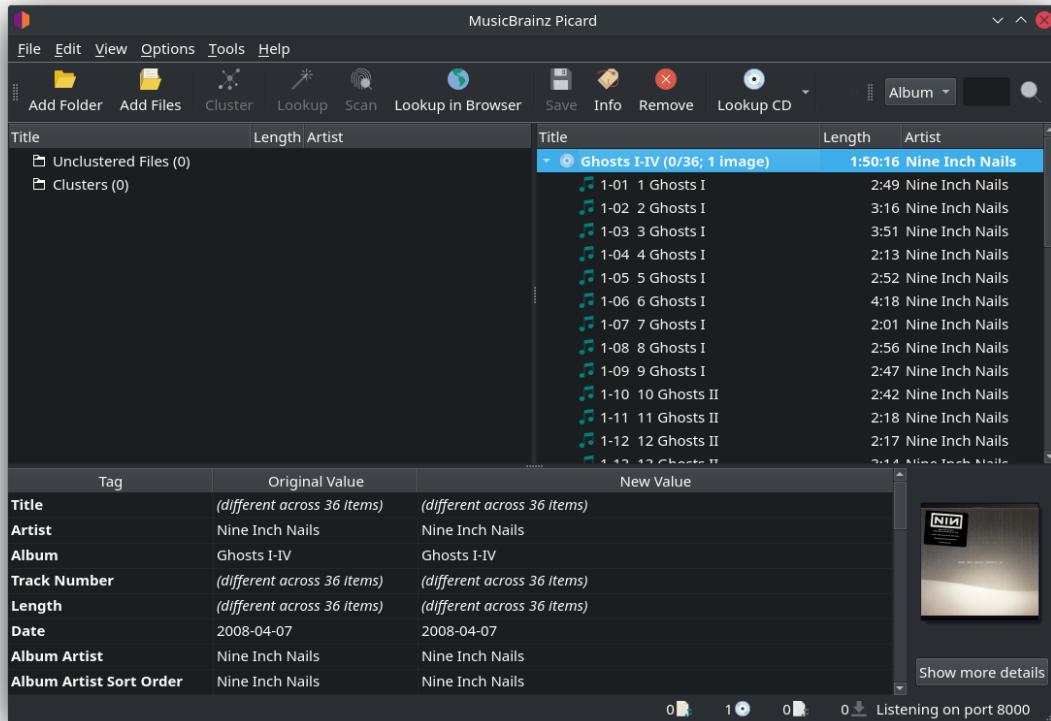
2. Scannez le code-barres d'un CD, d'un disque ou d'un autre support musical.



3. Si une version avec le code-barres scanné est trouvée sur MusicBrainz, l'application se chargera et montrera les détails de la version.



4. Tapez sur « Send to Picard ». Si tout a été configuré correctement, la version sera chargée dans Picard sur votre ordinateur.



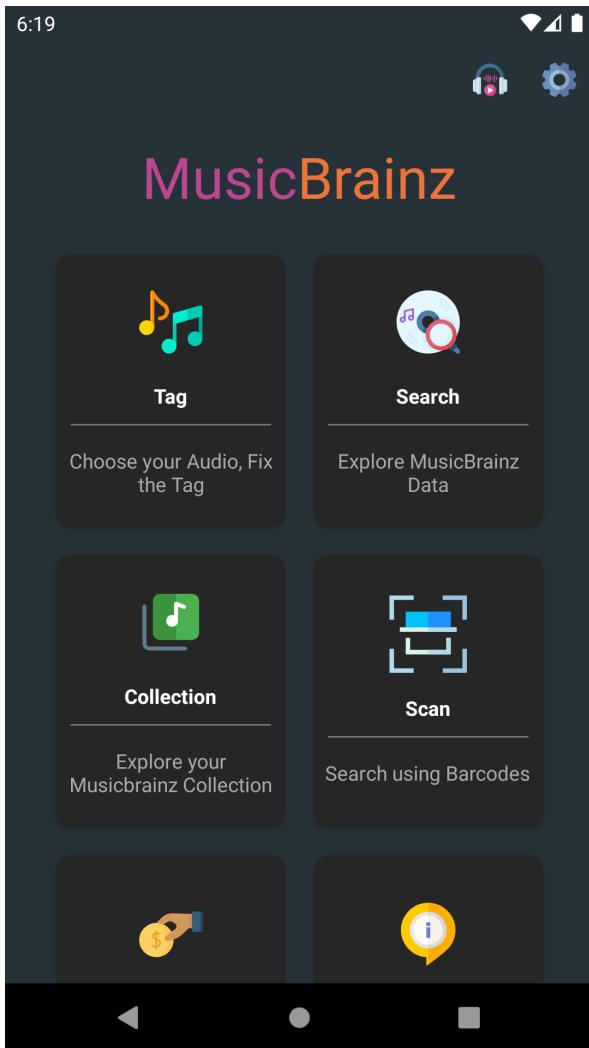
Vous pouvez maintenant continuer à étiqueter vos fichiers locaux en les faisant correspondre aux pistes chargées comme décrit dans [Mise en correspondance des fichiers aux pistes](#).

Note: Si vous souhaitez uniquement utiliser la fonctionnalité de scanner de codes barres pour trouver et charger les sorties de vos CD physiques, LP ou autres supports musicaux, vous pouvez également utiliser l'application [Picard Barcode Scanner](#). La fonctionnalité et la configuration sont similaires à ce qui est décrit ci-dessus, mais l'application se concentre sur la lecture des codes barres et l'envoi des résultats à Picard.

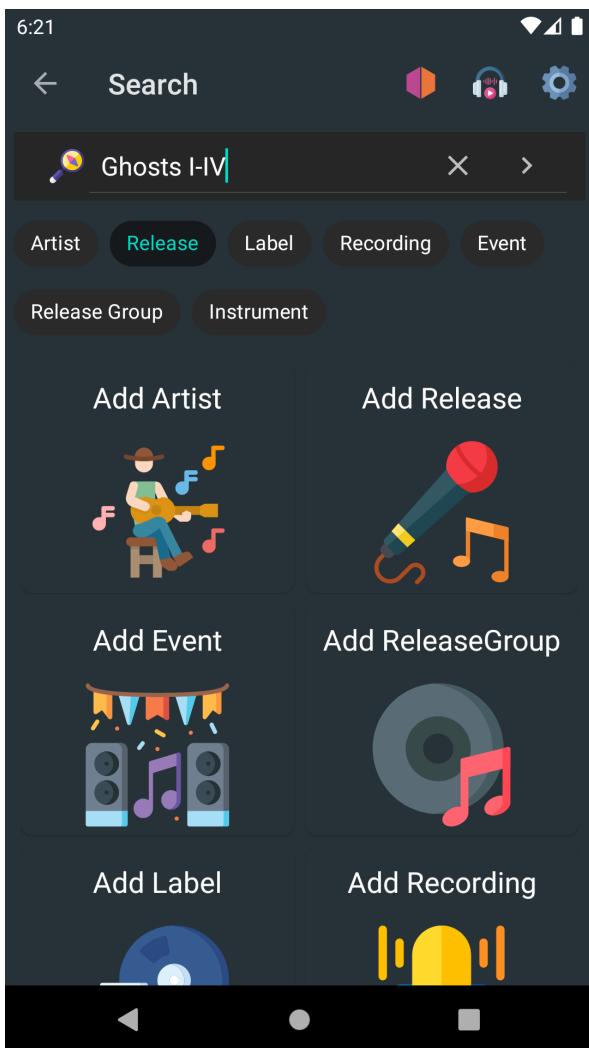
16.5.4 Chargement des communiqués par recherche

Au lieu d'effectuer une recherche par code-barres, vous pouvez également effectuer une recherche textuelle sur votre téléphone:

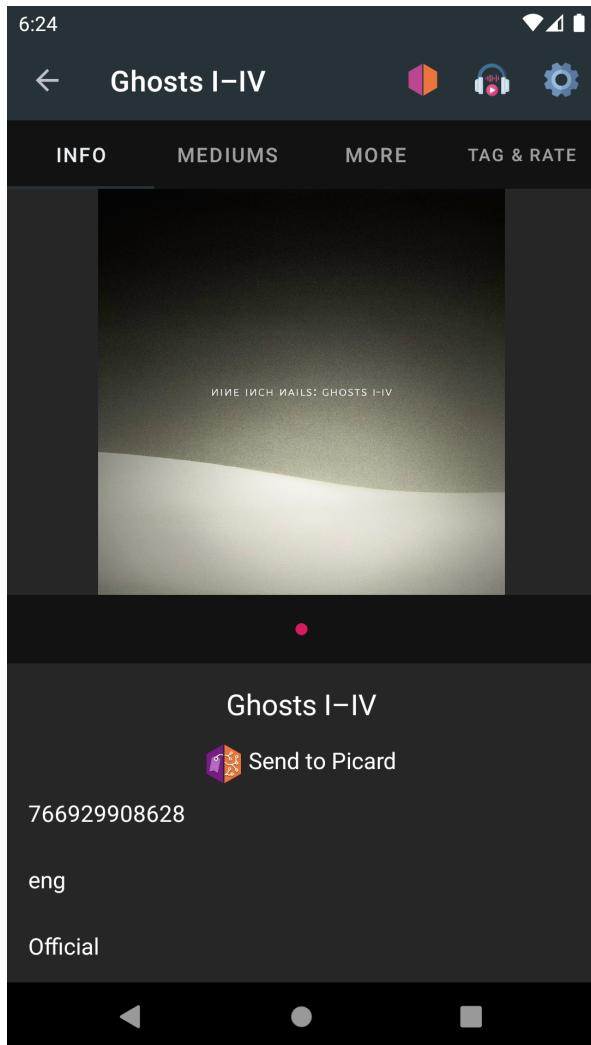
1. Sur l'écran principal de l'application Android, tapez sur « Recherche ».



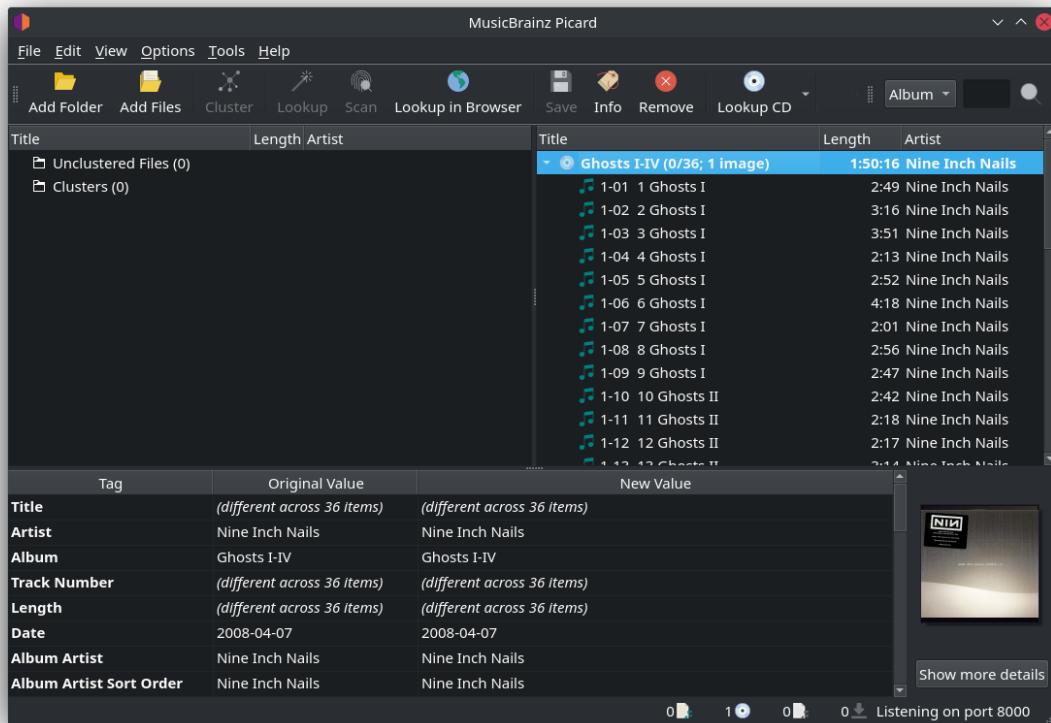
2. Sur la page de recherche, sélectionnez « Release » et entrez un terme de recherche, par exemple un titre d'album ou un nom d'artiste.



3. Les résultats de la recherche affichent une liste de communiqués correspondants. Tapez sur l'un d'entre eux pour afficher les détails du communiqué.



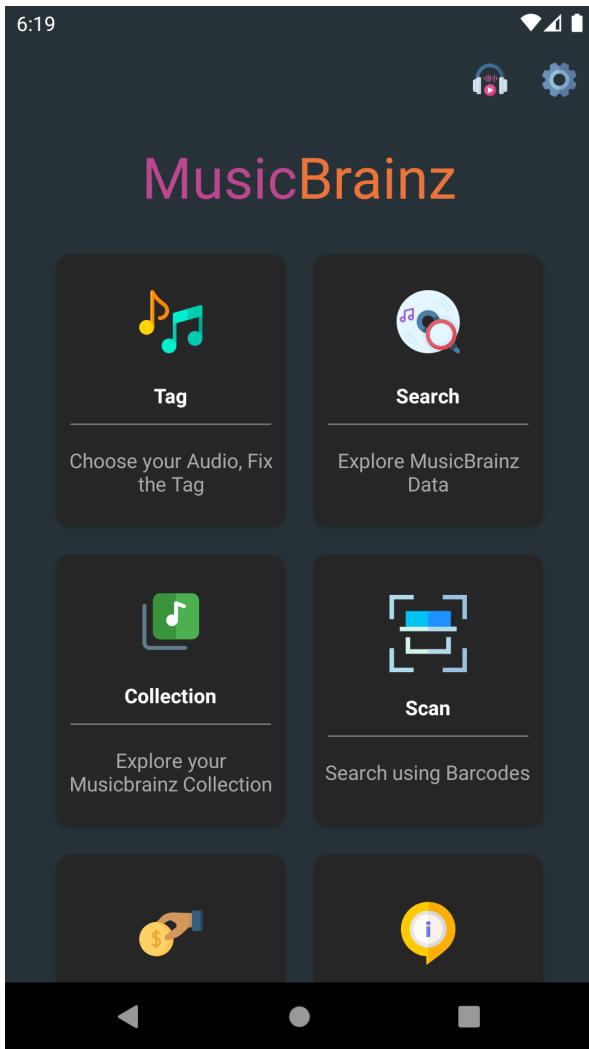
4. Tapez sur « Send to Picard ». Si tout a été configuré correctement, la version sera chargée dans Picard sur votre ordinateur.



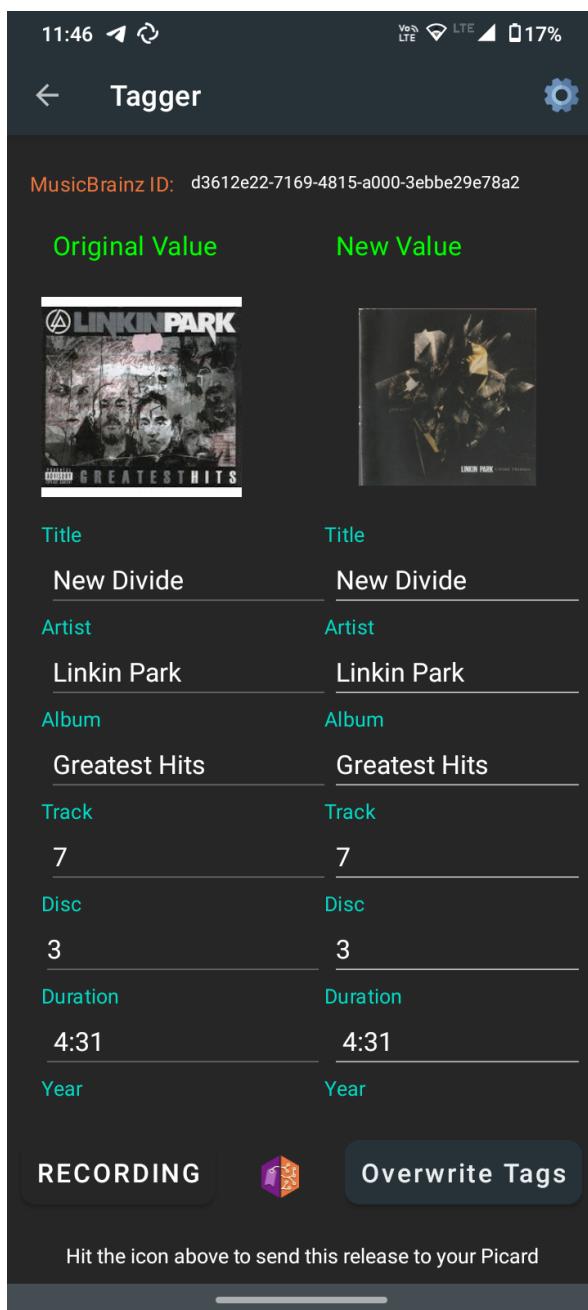
16.5.5 Chargement des communiqués du Tagger

Au lieu de trouver un communiqué par code-barres ou par une recherche pour l'envoyer à Picard, vous pouvez également envoyer un communiqué à partir d'un fichier audio balisé actuellement stocké sur votre appareil:

1. Sur l'écran principal de l'application Android, tapez sur « Tagger ».

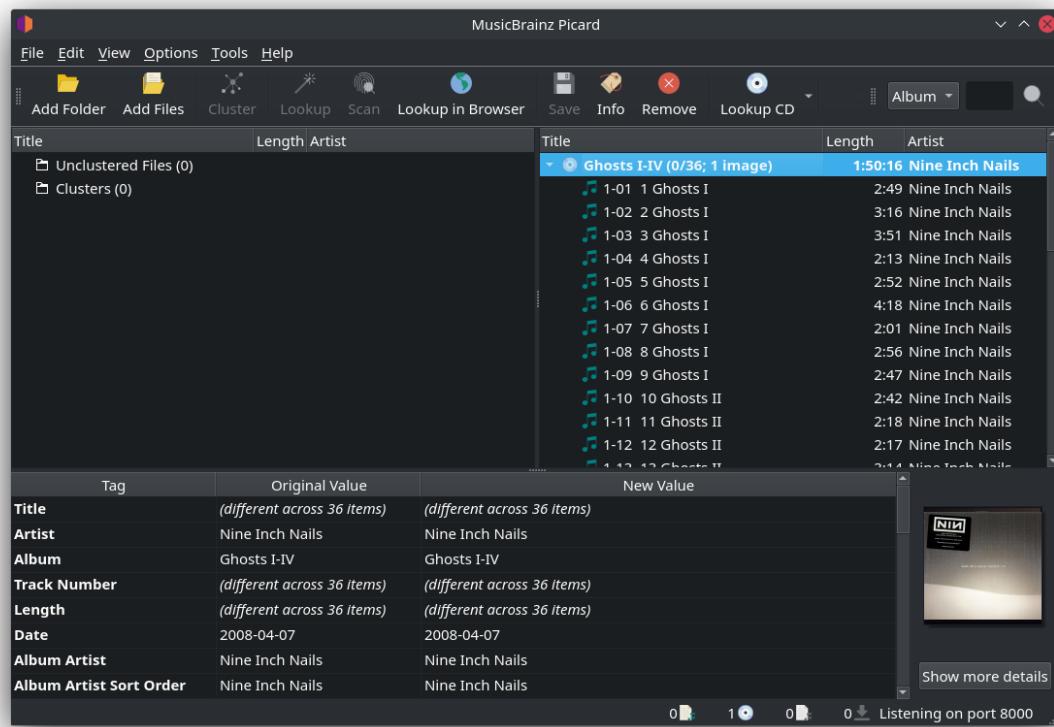


2. Sur le tagger, sélectionnez votre version et appuyez sur l'icône MusicBrainz en bas de l'écran.



- Si tout a été configuré correctement, la version sera chargée dans Picard qui tourne sur votre ordinateur.

MusicBrainz Picard, Version v2.6.3



Annexes

17.1 Annexe A: API des plugins

17.1.1 Métadonnées du plugin

Chaque plugin doit fournir des métadonnées sous forme de variables. Ces variables doivent être placées en haut du fichier.

```
PLUGIN_NAME = "Example plugin"
PLUGIN_AUTHOR = "This authors name"
PLUGIN_DESCRIPTION = "This plugin is an example"
PLUGIN_VERSION = '0.1'
PLUGIN_API VERSIONS = ['2.1', '2.2']
PLUGIN_LICENSE = "GPL-2.0-or-later"
PLUGIN_LICENSE_URL = "https://www.gnu.org/licenses/gpl-2.0.html"
```

Explication des variables:

- **PLUGIN_DESCRIPTION** doit être aussi simple que possible, tout en décrivant toujours la fonction principale.
- **PLUGIN_VERSION** doit être rempli avec la version de le Plug-in. Les versions de plug-in doivent être au format x.y.z (par exemple: « 1.0 » ou « 2.12.4 »). Il est recommandé d'utiliser [Semantic Versioning](#).
- **PLUGIN_API VERSIONS** doit être défini sur les versions de Picard avec lesquelles s'exécute ce plugin. Les nouvelles versions de Picard prendront généralement en charge les anciennes versions de l'API de plug-in, mais en cas de modification, la prise en charge des anciennes versions de plug-in peut être abandonnée. Les versions disponibles pour Picard 2 sont « 2.0 », « 2.1 » et « 2.2 ».

- **PLUGIN_LICENSE** doit être défini avec le nom de licence du plugin. Si possible, utilisez l'un des noms de licence de la [Liste des licences SPDX](#), mais vous êtes invités à utiliser une autre licence si celle que vous avez choisie n'est pas disponible dans la liste.
- **PLUGIN_LICENSE_URL** doit être défini sur une URL pointant vers le texte complet de la licence.

17.1.2 Processeurs de métadonnées

Les métadonnées MusicBrainz peuvent être post-traitées à deux niveaux, album et piste. Les types d'arguments passés aux fonctions de processeur dans les exemples suivants sont les suivants:

- **album**: picard.album.Album
- **metadata**: picard.metadata.Metadata
- **release**: dict avec les données de version du service Web MusicBrainz JSON
- **track**: dict avec les données de piste du service Web MusicBrainz JSON

Exemple de métadonnées d'album:

```
PLUGIN_NAME = "Disc Numbers"
PLUGIN_AUTHOR = "Lukas Lalinsky"
PLUGIN_DESCRIPTION = "Moves disc numbers from album titles to tags."

from picard.metadata import register_album_metadata_processor
import re

def remove_discnumbers(tagger, metadata, release):
    matches = re.search(r"\(disc (\d+)\)", metadata["album"])
    if matches:
        metadata["discnumber"] = matches.group(1)
        metadata["album"] = re.sub(r"\(disc \d+\)", "", metadata["album"])

register_album_metadata_processor(remove_discnumbers)
```

Exemple de Suivre les métadonnées:

```
PLUGIN_NAME = "Feat. Artists"
PLUGIN_AUTHOR = "Lukas Lalinsky"
PLUGIN_DESCRIPTION = "Removes feat. artists from track titles."

from picard.metadata import register_track_metadata_processor
import re

def remove_featartists(tagger, metadata, track, release):
```

(suite sur la page suivante)

(suite de la page précédente)

```
metadata["title"] = re.sub(r"\(feat. [^)]*\)", "", metadata["title"])
register_track_metadata_processor(remove_featartists)
```

17.1.3 Crochets d'événement

Les plugins peuvent s'enregistrer pour écouter différents événements. Actuellement, les hooks d'événement suivants sont disponibles:

file_post_load_processor(file)

Ce hook est appelé après le chargement d'un fichier dans Picard. Cela pourrait par exemple être utilisé pour charger des données supplémentaires pour un fichier. Usage:

```
from picard.file import register_file_post_load_processor

def file_post_load_processor(file):
    pass

register_file_post_load_processor(file_post_load_processor)
```

file_post_save_processor(file)

Ce hook est appelé après l'enregistrement d'un fichier. Cela peut par exemple être utilisé pour exécuter un post-traitement supplémentaire sur le fichier ou écrire des données supplémentaires. Notez que les métadonnées du fichier sont déjà les métadonnées nouvellement enregistrées. Usage:

```
from picard.file import register_file_post_save_processor

def file_post_save_processor(file):
    pass

register_file_post_save_processor(file_post_save_processor)
```

file_post_addition_to_track_processor(track, file)

Ce hook est appelé après qu'un fichier a été ajouté à une piste (dans le volet de droite de Picard).

```
from picard.file import register_file_post_addition_to_track_processor

def file_post_addition_to_track_processor(track, file):
    pass
```

(suite sur la page suivante)

(suite de la page précédente)

```
register_file_post_addition_to_track_processor(file_post_addition_to_track_
    ↪processor)
```

file_post_removal_from_track_processor(track, file)

Ce hook est appelé après qu'un fichier a été supprimé d'une piste (dans le volet droit de Picard).

```
from picard.file import register_file_post_removal_from_track_processor

def file_post_removal_from_track_processor(track, file):
    pass

register_file_post_removal_from_track_processor(file_post_removal_from_track_
    ↪processor)
```

album_post_removal_processor(album)

Ce hook est appelé après qu'un album a été supprimé de Picard.

```
from picard.album import register_album_post_removal_processor

def album_post_removal_processor(album):
    pass

register_album_post_removal_processor(album_post_removal_processor)
```

Note: Les hooks d'événement sont disponibles depuis la version 2.2 de l'API.

17.1.4 Formats de fichiers

Les plugins peuvent étendre Picard avec la prise en charge de formats de fichiers supplémentaires. Voir les [implémentations de format de fichier existantes](#) pour plus de détails sur la façon d'implémenter les méthodes « `_load` » et « `_save` ». Exemple:

```
PLUGIN_NAME = "..."
PLUGIN_AUTHOR = "..."
PLUGIN_DESCRIPTION = "..."
PLUGIN_VERSION = '...'
PLUGIN_API_VERSIONS = ['...']
PLUGIN_LICENSE = "..."
PLUGIN_LICENSE_URL = "..."
```

(suite sur la page suivante)

(suite de la page précédente)

```

from picard.file import File
from picard.formats import register_format
from picard.metadata import Metadata

class MyFile(File):
    EXTENSIONS = [".foo"]
    NAME = "Foo Audio"

    def _load(self, filename):
        metadata = Metadata()
        # Implement loading and parsing the file here.
        # This method is supposed to return a Metadata instance filled
        # with all the metadata read from the file.
        metadata['~format'] = self.NAME
        return metadata

    def _save(self, filename, metadata):
        # Implement saving the metadata to the file here.
        pass

register_format(MyFile)

```

17.1.5 Fonctions de script de tagger

Pour définir de nouvelles fonctions de script de tagger, utilisez `register_script_function(function, name = None)` du module `picard.script`. `parser` est une instance de `picard.script.ScriptParser`, et le reste des arguments qui lui sont passés sont les arguments de l'appel de fonction dans le script du tagger. Exemple:

```

PLUGIN_NAME = "Initials"
PLUGIN_AUTHOR = "Lukas Lalinsky"
PLUGIN_DESCRIPTION = "Provides tagger script function $initials(text)."
PLUGIN_VERSION = '0.1'
PLUGIN_API_VERSIONS = ['2.0']
PLUGIN_LICENSE = "GPL-2.0"
PLUGIN_LICENSE_URL = "https://www.gnu.org/licenses/gpl-2.0.txt"

from picard.script import register_script_function

def initials(parser, text):
    return "".join(a[:1] for a in text.split(" ") if a[:1].isalpha())

register_script_function(initials)

```

`register_script_function` prend en charge deux arguments optionnels:

- **eval_args**: Si c'est **False**, les arguments ne seront pas évalués avant d'être passés à **function**.

- **check_argcount**: Si c'est **False**, le nombre d'arguments passés à la fonction ne sera pas vérifié.

La valeur par défaut des deux arguments est **True**.

17.1.6 Actions du menu contextuel

Les actions du menu contextuel du clic droit peuvent être ajoutées aux albums, pistes et fichiers dans « Fichiers sans correspondance », « Clusters » et « ClusterList » (dossier parent des clusters). Exemple:

```
PLUGIN_NAME = u'Remove Perfect Albums'
PLUGIN_AUTHOR = u'ichneumon, hrglgrmpf'
PLUGIN_DESCRIPTION = u'''Remove all perfectly matched albums from the
↪selection.'''
PLUGIN_VERSION = '0.2'
PLUGIN_API_VERSIONS = ['0.15.1']
PLUGIN_LICENSE = "GPL-2.0"
PLUGIN_LICENSE_URL = "https://www.gnu.org/licenses/gpl-2.0.txt"

from picard.album import Album
from picard.ui.itemviews import BaseAction, register_album_action

class RemovePerfectAlbums(BaseAction):
    NAME = 'Remove perfect albums'

    def callback(self, objs):
        for album in objs:
            if isinstance(album, Album) and album.is_complete()\
                and album.get_num_unmatched_files() == 0\
                and album.get_num_matched_tracks() == len(list(album.
↪iterfiles()))\
                and album.get_num_unsaved_files() == 0 and album.loaded ==
↪True:
                self.tagger.remove_album(album)

register_album_action(RemovePerfectAlbums())
```

Utilisez `register_x_action` où “x” est « *album* », « *track* », « *file* », « *cluster* » ou « *clusterlist* ».

17.2 Annexe B: Mappage de balises

Ce qui suit est un mappage entre les noms de balises internes de Picard et ceux utilisés par divers formats de balisage. Le mappage est également disponible sous forme de [table](#) et de [tableur](#).

17.2.1 AcoustID

Nom interne	acoustid_id
ID3v2	TXXX:Acoustid Id
Vorbis	ACOUSTID_ID
APEv2	ACOUSTID_ID
iTunes MP4	----:com.apple.iTunes:Acoustid Id
ASF/Windows Media	Acoustid/Id
RIFF INFO	n/a

17.2.2 Empreinte digitale AcoustID

Nom interne	acoustid_fingerprint
ID3v2	TXXX:Acoustid Fingerprint
Vorbis	ACOUSTID_FINGERPRINT
APEv2	ACOUSTID_FINGERPRINT
iTunes MP4	----:com.apple.iTunes:Acoustid Fingerprint
ASF/Windows Media	Acoustid/Fingerprint
RIFF INFO	n/a

17.2.3 Album

Nom interne	album
ID3v2	TALB
Vorbis	ALBUM
APEv2	Album
iTunes MP4	©alb
ASF/Windows Media	WM/AlbumTitle
RIFF INFO	IPRD

17.2.4 Artiste de l'album

Nom interne	albumartist
ID3v2	TPE2
Vorbis	ALBUMARTIST
APEv2	Album Artist
iTunes MP4	aART
ASF/Windows Media	WM/AlbumArtist
RIFF INFO	n/a

17.2.5 Ordre de tri de l'artiste de l'album

Nom interne	albumartistsort
ID3v2	TS02 (Picard>=1.2) TXXX:ALBUMARTISTSORT (Picard<=1.1)
Vorbis	ALBUMARTISTSORT
APEv2	ALBUMARTISTSORT
iTunes MP4	soaa
ASF/Windows Media	WM/AlbumArtistSortOrder
RIFF INFO	n/a

17.2.6 Ordre de tri des albums ^[4]

Nom interne	albumsrt
ID3v2	TS0A
Vorbis	ALBUMSORT
APEv2	ALBUMSORT
iTunes MP4	soal
ASF/Windows Media	WM/AlbumSortOrder
RIFF INFO	n/a

17.2.7 Arrangeur

Nom interne	arranger
ID3v2	TIPL:arranger (ID3v2.4) IPLS:arranger (ID3v2.3)
Vorbis	ARRANGER
APEv2	Arranger
iTunes MP4	n/a
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.8 Artiste

Nom interne	artist
ID3v2	TPE1
Vorbis	ARTIST
APEv2	Artist
iTunes MP4	©ART
ASF/Windows Media	Author
RIFF INFO	IART

17.2.9 Ordre de tri des artistes

Nom interne	artistsort
ID3v2	TSOP
Vorbis	ARTISTSORT
APEv2	ARTISTSORT
iTunes MP4	soar
ASF/Windows Media	WM/ArtistSortOrder
RIFF INFO	n/a

17.2.10 Artistes

Nom interne	artists
ID3v2	TXXX:ARTISTS
Vorbis	ARTISTS
APEv2	Artists
iTunes MP4	----:com.apple.iTunes:ARTISTS
ASF/Windows Media	WM/ARTISTS
RIFF INFO	n/a

17.2.11 ASIN

Nom interne	asin
ID3v2	TXXX:ASIN
Vorbis	ASIN
APEv2	ASIN
iTunes MP4	----:com.apple.iTunes:ASIN
ASF/Windows Media	ASIN
RIFF INFO	n/a

17.2.12 Code à barres

Nom interne	barcode
ID3v2	TXXX:BARCODE
Vorbis	BARCODE
APEv2	Barcode
iTunes MP4	----:com.apple.iTunes:BARCODE
ASF/Windows Media	WM/Barcode
RIFF INFO	n/a

17.2.13 BPM ^[4]

Nom interne	bpm
ID3v2	TBPM
Vorbis	BPM
APEv2	BPM
iTunes MP4	tempo
ASF/Windows Media	WM/BeatsPerMinute
RIFF INFO	n/a

17.2.14 Numéro de catalogue

Nom interne	catalognumber
ID3v2	TXXX:CATALOGNUMBER
Vorbis	CATALOGNUMBER
APEv2	CatalogNumber
iTunes MP4	----:com.apple.iTunes:CATALOGNUMBER
ASF/Windows Media	WM/CatalogNo
RIFF INFO	n/a

17.2.15 Commentaire ^[4]

Nom interne	comment:description
ID3v2	COMM:description
Vorbis	COMMENT
APEv2	Comment
iTunes MP4	©cmt
ASF/Windows Media	Description
RIFF INFO	ICMT

17.2.16 Compilation (iTunes) [5]

Nom interne	compilation
ID3v2	TCMP
Vorbis	COMPILATION
APEv2	Compilation
iTunes MP4	cpil
ASF/Windows Media	WM/IsCompilation
RIFF INFO	n/a

17.2.17 Compositeur

Nom interne	composer
ID3v2	TCOM
Vorbis	COMPOSER
APEv2	Composer
iTunes MP4	©wrt
ASF/Windows Media	WM/Composer
RIFF INFO	IMUS

17.2.18 Ordre de tri du compositeur

Nom interne	composersort
ID3v2	TSOC (Picard>=1.3) TXXX:COMPOSERSORT (Picard<=1.2)
Vorbis	COMPOSERSORT
APEv2	COMPOSERSORT
iTunes MP4	soco
ASF/Windows Media	WM/ComposerSortOrder (Picard>=1.3)
RIFF INFO	n/a

17.2.19 Chef d'orchestre

Nom interne	conductor
ID3v2	TPE3
Vorbis	CONDUCTOR
APEv2	Conductor
iTunes MP4	----:com.apple.iTunes:CONDUCTOR
ASF/Windows Media	WM/Conductor
RIFF INFO	n/a

17.2.20 Copyright ^[4]

Nom interne	copyright
ID3v2	TCOP
Vorbis	COPYRIGHT
APEv2	Copyright
iTunes MP4	cprt
ASF/Windows Media	Copyright
RIFF INFO	ICOP

17.2.21 Réalisateur

Nom interne	director
ID3v2	TXXX:DIRECTOR
Vorbis	DIRECTOR
APEv2	Director
iTunes MP4	mdir ^[9]
ASF/Windows Media	WM/Director
RIFF INFO	n/a

17.2.22 Numéro de disque

Nom interne	discnumber
ID3v2	TP0S
Vorbis	DISCNUMBER
APEv2	Disc
iTunes MP4	disk
ASF/Windows Media	WM/PartOfSet
RIFF INFO	n/a

17.2.23 Sous-titre du disque

Nom interne	discsubtitle
ID3v2	TSST (ID3v2.4 only)
Vorbis	DISCSUBTITLE
APEv2	DiscSubtitle
iTunes MP4	----:com.apple.iTunes:DISCSUBTITLE
ASF/Windows Media	WM/SetSubTitle
RIFF INFO	n/a

17.2.24 Encodé par [4]

Nom interne	encodedby
ID3v2	TENC
Vorbis	ENCODEDBY
APEv2	EncodedBy
iTunes MP4	©too
ASF/Windows Media	WM/EncodedBy
RIFF INFO	IENC

17.2.25 Paramètres de l'encodeur [4]

Nom interne	encodersettings
ID3v2	TSSE
Vorbis	ENCODERSETTINGS
APEv2	EncoderSettings
iTunes MP4	n/a
ASF/Windows Media	WM/EncodingSettings (Picard>=1.3.1)
RIFF INFO	n/a

17.2.26 Ingénieur

Nom interne	engineer
ID3v2	TIPL:engineer (ID3v2.4) IPLS:engineer (ID3v2.3)
Vorbis	ENGINEER
APEv2	Engineer
iTunes MP4	- - - :com.apple.iTunes:ENGINEER
ASF/Windows Media	WM/Engineer
RIFF INFO	IENG

17.2.27 Lecture sans interruption [4]

Nom interne	gapless
ID3v2	n/a
Vorbis	n/a
APEv2	n/a
iTunes MP4	pgap
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.28 Genre

Nom interne	genre
ID3v2	TCON
Vorbis	GENRE
APEv2	Genre
iTunes MP4	©gen
ASF/Windows Media	WM/Genre
RIFF INFO	IGNR

17.2.29 Regroupement [3]

Nom interne	grouping
ID3v2	TIT1 GRP1 [8]
Vorbis	GROUPING
APEv2	Grouping
iTunes MP4	©grp
ASF/Windows Media	WM/ContentGroupDescription
RIFF INFO	n/a

17.2.30 Clé initiale

Nom interne	key (Picard>=1.4)
ID3v2	TKEY
Vorbis	KEY
APEv2	KEY
iTunes MP4	----:com.apple.iTunes:initialkey
ASF/Windows Media	WM/InitialKey
RIFF INFO	n/a

17.2.31 ISRC

Nom interne	isrc
ID3v2	TSRC
Vorbis	ISRC
APEv2	ISRC
iTunes MP4	----:com.apple.iTunes:ISRC
ASF/Windows Media	WM/ISRC
RIFF INFO	n/a

17.2.32 Langue

Nom interne	language
ID3v2	TLAN
Vorbis	LANGUAGE
APEv2	Language
iTunes MP4	----:com.apple.iTunes:LANGUAGE
ASF/Windows Media	WM/Language
RIFF INFO	ILNG

17.2.33 License ^[6, 7]

Nom interne	license
ID3v2	WCOP (single URL) TXXX:LICENSE (multiple or non-URL)
Vorbis	LICENSE
APEv2	LICENSE
iTunes MP4	----:com.apple.iTunes:LICENSE
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.34 Parolier

Nom interne	lyricist
ID3v2	TEXT
Vorbis	LYRICIST
APEv2	Lyricist
iTunes MP4	----:com.apple.iTunes:LYRICIST
ASF/Windows Media	WM/Writer
RIFF INFO	n/a

17.2.35 Paroles ^[4]

Nom interne	lyrics:description
ID3v2	USLT:description
Vorbis	LYRICS
APEv2	Lyrics
iTunes MP4	©lyr
ASF/Windows Media	WM/Lyrics
RIFF INFO	n/a

17.2.36 Médias

Nom interne	media
ID3v2	TMED
Vorbis	MEDIA
APEv2	Media
iTunes MP4	----:com.apple.iTunes:MEDIA
ASF/Windows Media	WM/Media
RIFF INFO	IMED

17.2.37 Mix-DJ

Nom interne	djmixer
ID3v2	TIPL:DJ-mix (ID3v2.4) IPLS:DJ-mix (ID3v2.3)
Vorbis	DJMIXER
APEv2	DJMixer
iTunes MP4	----:com.apple.iTunes:DJMIXER
ASF/Windows Media	WM/DJMixer
RIFF INFO	n/a

17.2.38 Mélangeur

Nom interne	mixer
ID3v2	TIPL:mix (ID3v2.4) IPLS:mix (ID3v2.3)
Vorbis	MIXER
APEv2	Mixer
iTunes MP4	----:com.apple.iTunes:MIXER
ASF/Windows Media	WM/Mixer
RIFF INFO	n/a

17.2.39 Mood [3]

Nom interne	mood
ID3v2	TM00 (ID3v2.4 only)
Vorbis	MOOD
APEv2	Mood
iTunes MP4	----:com.apple.iTunes:MOOD
ASF/Windows Media	WM/Mood
RIFF INFO	n/a

17.2.40 Mouvement [4]

Nom interne	movement (Picard>=2.1)
ID3v2	MVNM
Vorbis	MOVEMENTNAME
APEv2	MOVEMENTNAME
iTunes MP4	©mvn
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.41 Nombre de mouvements [4]

Nom interne	movementtotal (Picard>=2.1)
ID3v2	MVIN
Vorbis	MOVEMENTTOTAL
APEv2	MOVEMENTTOTAL
iTunes MP4	mvc
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.42 Numéro de mouvement [4]

Nom interne	movementnumber (Picard>=2.1)
ID3v2	MVIN
Vorbis	MOVEMENT
APEv2	MOVEMENT
iTunes MP4	mvi
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.43 Identifiant de l'artiste MusicBrainz

Nom interne	musicbrainz_artistid
ID3v2	TXXX:MusicBrainz Artist Id
Vorbis	MUSICBRAINZ_ARTISTID
APEv2	MUSICBRAINZ_ARTISTID
iTunes MP4	--:com.apple.iTunes:MusicBrainz Artist Id
ASF/Windows Media	MusicBrainz/Artist Id
RIFF INFO	n/a

17.2.44 ID du disque MusicBrainz

Nom interne	musicbrainz_discid
ID3v2	TXXX:MusicBrainz Disc Id
Vorbis	MUSICBRAINZ_DISCID
APEv2	MUSICBRAINZ_DISCID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Disc Id
ASF/Windows Media	MusicBrainz/Disc Id
RIFF INFO	n/a

17.2.45 Identifiant original de l'artiste MusicBrainz

Nom interne	musicbrainz_originalartistid
ID3v2	TXXX:MusicBrainz Original Artist Id
Vorbis	MUSICBRAINZ_ORIGINALARTISTID
APEv2	n/a
iTunes MP4	----:com.apple.iTunes:MusicBrainz Original Artist Id (Picard>=2.1)
ASF/Windows Media	MusicBrainz/Original Artist Id (Picard>=2.1)
RIFF INFO	n/a

17.2.46 ID de la version originale de MusicBrainz

Nom interne	musicbrainz_originalalbumid
ID3v2	TXXX:MusicBrainz Original Album Id
Vorbis	MUSICBRAINZ_ORIGINALALBUMID
APEv2	n/a
iTunes MP4	----:com.apple.iTunes:MusicBrainz Original Album Id (Picard>=2.1)
ASF/Windows Media	MusicBrainz/Original Album Id (Picard>=2.1)
RIFF INFO	n/a

17.2.47 Identifiant d'enregistrement MusicBrainz

Nom interne	musicbrainz_recordingid
ID3v2	UFID:http://musicbrainz.org
Vorbis	MUSICBRAINZ_TRACKID
APEv2	MUSICBRAINZ_TRACKID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Track Id
ASF/Windows Media	MusicBrainz/Track Id
RIFF INFO	n/a

17.2.48 Identifiant de l'artiste de la version MusicBrainz

Nom interne	musicbrainz_albumartistid
ID3v2	TXXX:MusicBrainz Album Artist Id
Vorbis	MUSICBRAINZ_ALBUMARTISTID
APEv2	MUSICBRAINZ_ALBUMARTISTID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Album Artist Id
ASF/Windows Media	MusicBrainz/Album Artist Id
RIFF INFO	n/a

17.2.49 Identifiant du groupe de versions MusicBrainz

Nom interne	musicbrainz_releasegroupid
ID3v2	TXXX:MusicBrainz Release Group Id
Vorbis	MUSICBRAINZ_RELEASEGROUPID
APEv2	MUSICBRAINZ_RELEASEGROUPID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Release Group Id
ASF/Windows Media	MusicBrainz/Release Group Id
RIFF INFO	n/a

17.2.50 Identifiant de version MusicBrainz

Nom interne	musicbrainz_albumid
ID3v2	TXXX:MusicBrainz Album Id
Vorbis	MUSICBRAINZ_ALBUMID
APEv2	MUSICBRAINZ_ALBUMID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Album Id
ASF/Windows Media	MusicBrainz/Album Id
RIFF INFO	n/a

17.2.51 ID de la piste MusicBrainz

Nom interne	musicbrainz_trackid
ID3v2	TXXX:MusicBrainz Release Track Id
Vorbis	MUSICBRAINZ_RELEASETRACKID
APEv2	MUSICBRAINZ_RELEASETRACKID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Release Track Id
ASF/Windows Media	MusicBrainz/Release Track Id
RIFF INFO	n/a

17.2.52 MusicBrainz TRM ID

Nom interne	musicbrainz_trmid (deprecated)
ID3v2	TXXX:MusicBrainz TRM Id
Vorbis	MUSICBRAINZ_TRMID
APEv2	MUSICBRAINZ_TRMID
iTunes MP4	----:com.apple.iTunes:MusicBrainz TRM Id
ASF/Windows Media	MusicBrainz/TRM Id
RIFF INFO	n/a

17.2.53 MusicBrainz Work ID

Nom interne	musicbrainz_workid
ID3v2	TXXX:MusicBrainz Work Id
Vorbis	MUSICBRAINZ_WORKID
APEv2	MUSICBRAINZ_WORKID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Work Id
ASF/Windows Media	MusicBrainz/Work Id
RIFF INFO	n/a

17.2.54 Empreinte digitale MusicIP

Nom interne	musicip_fingerprint
ID3v2	TXXX:MusicMagic Fingerprint
Vorbis	FINGERPRINT=MusicMagic Fingerprint {fingerprint}
APEv2	n/a
iTunes MP4	----:com.apple.iTunes:fingerprint
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.55 MusicIP PUID

Nom interne	musicip_puid
ID3v2	TXXX:MusicIP PUID
Vorbis	MUSICIP_PUID
APEv2	MUSICIP_PUID
iTunes MP4	----:com.apple.iTunes:MusicIP PUID
ASF/Windows Media	MusicIP/PUID
RIFF INFO	n/a

17.2.56 Album original

Nom interne	originalalbum
ID3v2	TOAL
Vorbis	n/a
APEv2	n/a
iTunes MP4	n/a
ASF/Windows Media	WM/OriginalAlbumTitle (Picard>=2.1)
RIFF INFO	n/a

17.2.57 Artiste original

Nom interne	originalartist
ID3v2	TOPE
Vorbis	n/a
APEv2	Original Artist (Picard>=2.1)
iTunes MP4	n/a
ASF/Windows Media	WM/OriginalArtist (Picard>=2.1)
RIFF INFO	n/a

17.2.58 Nom de fichier d'origine

Nom interne	originalfilename (Picard>=2.3)
ID3v2	TOFN
Vorbis	ORIGINALFILENAME
APEv2	ORIGINALFILENAME
iTunes MP4	n/a
ASF/Windows Media	WM/OriginalFilename
RIFF INFO	n/a

17.2.59 Date de sortie d'origine [1]

Nom interne	originaldate
ID3v2	TDOR (ID3v2.4) TORY (ID3v2.3)
Vorbis	ORIGINALDATE
APEv2	n/a
iTunes MP4	n/a
ASF/Windows Media	WM/OriginalReleaseTime (Picard>=1.3.1) WM/OriginalReleaseYear (Picard<=1.3.0)
RIFF INFO	n/a

17.2.60 Année de sortie d'origine ^[1]

Nom interne	originalyear
ID3v2	n/a
Vorbis	ORIGINALYEAR
APEv2	ORIGINALYEAR
iTunes MP4	n/a
ASF/Windows Media	WM/OriginalReleaseYear (Picard>=1.3.1)
RIFF INFO	n/a

17.2.61 L'exécutant

Nom interne	performer:instrument
ID3v2	TMCL:instrument (ID3v2.4) IPLS:instrument (ID3v2.3)
Vorbis	PERFORMER={artist} (instrument)
APEv2	Performer={artist} (instrument)
iTunes MP4	n/a
ASF/Windows Media	n/a
RIFF INFO	n/a

Voir aussi:

Veuillez vous référer à [Types de relations / Artiste-Release / Performer, Relation Types / Artist-Release / Vocal, Relation Types / Artist-Release / Instrument, Types de relations / Enregistrement d'artiste / Interprète, Types de relations / Enregistrement d'artiste / Vocal, et Relationship Types / Artist-Recording / Instrument](#) pour plus d'informations.

17.2.62 Podcast ^[4]

Nom interne	podcast
ID3v2	n/a
Vorbis	n/a
APEv2	n/a
iTunes MP4	pcst
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.63 Podcast URL [4]

Nom interne	podcasturl
ID3v2	n/a
Vorbis	n/a
APEv2	n/a
iTunes MP4	purl
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.64 Producteur

Nom interne	producer
ID3v2	TIPL:producer (ID3v2.4) IPLS:producer (ID3v2.3)
Vorbis	PRODUCER
APEv2	Producer
iTunes MP4	----:com.apple.iTunes:PRODUCER
ASF/Windows Media	WM/Producer
RIFF INFO	IPRO

17.2.65 Évaluation

Nom interne	_rating
ID3v2	POPM
Vorbis	RATING:user@email
APEv2	n/a
iTunes MP4	n/a
ASF/Windows Media	WM/SharedUserRating
RIFF INFO	n/a

17.2.66 Étiquette d'enregistrement

Nom interne	label
ID3v2	TPUB
Vorbis	LABEL
APEv2	Label
iTunes MP4	----:com.apple.iTunes:LABEL
ASF/Windows Media	WM/Publisher
RIFF INFO	n/a

17.2.67 Pays de sortie

Nom interne	releasecountry
ID3v2	TXXX:MusicBrainz Album Release Country
Vorbis	RELEASECOUNTRY
APEv2	RELEASECOUNTRY
iTunes MP4	----:com.apple.iTunes:MusicBrainz Album Release Country
ASF/Windows Media	MusicBrainz/Album Release Country
RIFF INFO	ICNT

17.2.68 Date de sortie

Nom interne	date
ID3v2	TDRC (ID3v2.4) TYER + TDAT (ID3v2.3)
Vorbis	DATE
APEv2	Year
iTunes MP4	©day
ASF/Windows Media	WM/Year
RIFF INFO	ICRD

17.2.69 État de sortie

Nom interne	releasestatus
ID3v2	TXXX:MusicBrainz Album Status
Vorbis	RELEASESTATUS
APEv2	MUSICBRAINZ_ALBUMSTATUS
iTunes MP4	----:com.apple.iTunes:MusicBrainz Album Status
ASF/Windows Media	MusicBrainz/Album Status
RIFF INFO	n/a

17.2.70 Type de version

Nom interne	releasetype
ID3v2	TXXX:MusicBrainz Album Type
Vorbis	RELEASETYPE
APEv2	MUSICBRAINZ_ALBUMTYPE
iTunes MP4	----:com.apple.iTunes:MusicBrainz Album Type
ASF/Windows Media	MusicBrainz/Album Type
RIFF INFO	n/a

17.2.71 Remixer

Nom interne	remixer
ID3v2	TPE4
Vorbis	REMIXER
APEv2	MixArtist
iTunes MP4	----:com.apple.iTunes:REMIXER
ASF/Windows Media	WM/ModifiedBy
RIFF INFO	n/a

17.2.72 ReplayGain Gain d'album

Nom interne	replaygain_album_gain (Picard>=2.2)
ID3v2	TXXX:REPLAYGAIN_ALBUM_GAIN
Vorbis	REPLAYGAIN_ALBUM_GAIN
APEv2	REPLAYGAIN_ALBUM_GAIN
iTunes MP4	----:com.apple.iTunes:REPLAYGAIN_ALBUM_GAIN
ASF/Windows Media	REPLAYGAIN_ALBUM_GAIN
RIFF INFO	n/a

17.2.73 ReplayGain Album Peak

Nom interne	replaygain_album_peak (Picard>=2.2)
ID3v2	TXXX:REPLAYGAIN_ALBUM_PEAK
Vorbis	REPLAYGAIN_ALBUM_PEAK
APEv2	REPLAYGAIN_ALBUM_PEAK
iTunes MP4	----:com.apple.iTunes:REPLAYGAIN_ALBUM_PEAK
ASF/Windows Media	REPLAYGAIN_ALBUM_PEAK
RIFF INFO	n/a

17.2.74 ReplayGain Album Range

Nom interne	replaygain_album_range (Picard>=2.2)
ID3v2	TXXX:REPLAYGAIN_ALBUM_RANGE
Vorbis	REPLAYGAIN_ALBUM_RANGE
APEv2	REPLAYGAIN_ALBUM_RANGE
iTunes MP4	----:com.apple.iTunes:REPLAYGAIN_ALBUM_RANGE
ASF/Windows Media	REPLAYGAIN_ALBUM_RANGE
RIFF INFO	n/a

17.2.75 ReplayGain Reference Loudness

Nom interne	replaygain_reference_loudness (Picard>=2.2)
ID3v2	TXXX:REPLAYGAIN_REFERENCELOUDNESS
Vorbis	REPLAYGAIN_REFERENCELOUDNESS
APEv2	REPLAYGAIN_REFERENCELOUDNESS
iTunes MP4	- - - :com.apple.iTunes:REPLAYGAIN_REFERENCELOUDNESS
ASF/Windows Media	REPLAYGAIN_REFERENCELOUDNESS
RIFF INFO	n/a

17.2.76 ReplayGain Track Gain

Nom interne	replaygain_track_gain (Picard>=2.2)
ID3v2	TXXX:REPLAYGAIN_TRACK_GAIN
Vorbis	REPLAYGAIN_TRACK_GAIN
APEv2	REPLAYGAIN_TRACK_GAIN
iTunes MP4	- - - :com.apple.iTunes:REPLAYGAIN_TRACK_GAIN
ASF/Windows Media	REPLAYGAIN_TRACK_GAIN
RIFF INFO	n/a

17.2.77 ReplayGain Track Peak

Nom interne	replaygain_track_peak (Picard>=2.2)
ID3v2	TXXX:REPLAYGAIN_TRACK_PEAK
Vorbis	REPLAYGAIN_TRACK_PEAK
APEv2	REPLAYGAIN_TRACK_PEAK
iTunes MP4	- - - :com.apple.iTunes:REPLAYGAIN_TRACK_PEAK
ASF/Windows Media	REPLAYGAIN_TRACK_PEAK
RIFF INFO	n/a

17.2.78 ReplayGain Track Range

Nom interne	replaygain_track_range (Picard>=2.2)
ID3v2	TXXX:REPLAYGAIN_TRACK_RANGE
Vorbis	REPLAYGAIN_TRACK_RANGE
APEv2	REPLAYGAIN_TRACK_RANGE
iTunes MP4	- - - :com.apple.iTunes:REPLAYGAIN_TRACK_RANGE
ASF/Windows Media	REPLAYGAIN_TRACK_RANGE
RIFF INFO	n/a

17.2.79 Script

Nom interne	script
ID3v2	TXXX:SCRIPT
Vorbis	SCRIPT
APEv2	Script
iTunes MP4	----:com.apple.iTunes:SCRIPT
ASF/Windows Media	WM/Script
RIFF INFO	n/a

17.2.80 Afficher le nom ^[4]

Nom interne	show
ID3v2	n/a
Vorbis	n/a
APEv2	n/a
iTunes MP4	tvsh
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.81 Afficher l'ordre de tri des noms ^[4]

Nom interne	showsrt
ID3v2	n/a
Vorbis	n/a
APEv2	n/a
iTunes MP4	sosn
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.82 Afficher le travail et le mouvement ^[4]

Nom interne	showmovement (Picard>=2.1)
ID3v2	TXXX:SHOWMOVEMENT
Vorbis	SHOWMOVEMENT
APEv2	SHOWMOVEMENT
iTunes MP4	shwm
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.83 Sous-titre [4]

Nom interne	subtitle
ID3v2	TIT3
Vorbis	SUBTITLE
APEv2	Subtitle
iTunes MP4	----:com.apple.iTunes:SUBTITLE
ASF/Windows Media	WM/SubTitle
RIFF INFO	n/a

17.2.84 Nombre total de disques

Nom interne	totaldiscs
ID3v2	TPOS
Vorbis	DISCTOTAL and TOTALDISCS
APEv2	Disc
iTunes MP4	disk
ASF/Windows Media	WM/Part0fSet (Picard>=1.3.1)
RIFF INFO	n/a

17.2.85 Total des pistes

Nom interne	totaltracks
ID3v2	TRCK
Vorbis	TRACKTOTAL and TOTALTRACKS
APEv2	Track
iTunes MP4	trkn
ASF/Windows Media	n/a
RIFF INFO	n/a

17.2.86 Numéro de piste

Nom interne	tracknumber
ID3v2	TRCK
Vorbis	TRACKNUMBER
APEv2	Track
iTunes MP4	trkn
ASF/Windows Media	WM/TrackNumber
RIFF INFO	ITRK

17.2.87 Titre de la piste

Nom interne	title
ID3v2	TIT2
Vorbis	TITLE
APEv2	Title
iTunes MP4	©nam
ASF/Windows Media	Title
RIFF INFO	INAM

17.2.88 Ordre de tri des titres de piste [4]

Nom interne	titlesort
ID3v2	TSOT
Vorbis	TITLESORT
APEv2	TITLESORT
iTunes MP4	sonm
ASF/Windows Media	WM/TitleSortOrder
RIFF INFO	n/a

17.2.89 Site Web (site officiel de l'artiste)

Nom interne	website
ID3v2	WOAR
Vorbis	WEBSITE
APEv2	Weblink
iTunes MP4	n/a
ASF/Windows Media	WM/AuthorURL (Picard>=1.3.1)
RIFF INFO	n/a

17.2.90 Work Title

Nom interne	work (Picard>=1.3)
ID3v2	TXXX:WORK TIT1 [8]
Vorbis	WORK
APEv2	WORK
iTunes MP4	©wrk (Picard>=2.1)
ASF/Windows Media	WM/Work
RIFF INFO	n/a

17.2.91 Écrivain [2]

Nom interne	writer
ID3v2	TXXX:Writer (Picard>=1.3)
Vorbis	WRITER
APEv2	Writer
iTunes MP4	n/a
ASF/Windows Media	n/a
RIFF INFO	IWRI

Remarques:

1. Tiré de la version la plus ancienne du groupe de versions.
2. Utilisé en cas de doute quant à savoir s'il s'agit d'un compositeur ou d'un parolier.
3. Ceci est fourni par le plugin LastFMPlus et non par Picard.
4. Ceci ne peut pas être fourni par Picard. Il peut être utilisé et rempli par certains plugins.
5. Pour Picard >= 1.3, cela indique un album de Various Artists; pour Picard <= 1.2, cela indique des albums avec des pistes d'artistes différents ce qui est incorrect (par exemple: un album original avec un duo avec un artiste de l'exploit apparaîtrait comme une compilation). Dans aucun des cas, cela n'indique un sous-type MusicBrainz Release Group de compilation.
6. [Type de relation de licence au niveau de la version](#).
7. [Type de relation de licence au niveau de l'enregistrement](#).
8. Avec « Enregistrer le regroupement compatible iTunes et travailler » (depuis Picard >= 2.1.0)
9. À partir de la spécification du format des métadonnées iTunes

17.3 Annexe C: Options de ligne de commande

Picard peut être démarré à partir de la ligne de commande avec les arguments suivants:

```
picard [-h] [-c CONFIG_FILE] [-d] [-N] [-P] [-v] [-V] [FILE [FILE ...]]
```

où les options sont:

- h, --help**
affiche un message d'aide et quitte
- c CONFIG_FILE, --config-file CONFIG_FILE**
emplacement du fichier de configuration à utiliser

-d, --debug

active la journalisation au niveau du débogage

-M, --no-player

désactiver le lecteur multimédia intégré

-N, --no-restore

ne restaure pas la position ou la taille des fenêtres

-P, --no-plugins

ne chargent aucun plug-in

-v, --version

affiche les informations de version et quitte

-V, --long-version

affiche les informations de la version longue et quitte

FILE

le ou les fichiers à charger

17.4 Annexe D: Courts clavier

17.4.1 Fenêtre principale

Fichier

Action	Windows / Linux	macOS
Ajouter un dossier	Ctrl+E	⌘+E
Ajouter des fichiers	Ctrl+O	⌘+O
Sauvegarder les fichiers sélectionnés	Ctrl+S	⌘+S
Quitter Picard	Ctrl+Q	⌘+Q

Modifier

Action	Windows / Linux	macOS
Couper les fichiers sélectionnés	Ctrl+X	⌘+X
Coller les fichiers sélectionnés	Ctrl+V	⌘+V
Afficher les informations sur l'élément sélectionné	Ctrl+I	⌘+I

Voir

Action	Windows / Linux	macOS
Basculer le navigateur de fichiers	Ctrl+B	⌘+B

Outils

Action	Windows / Linux	macOS
Rafraîchir	Ctrl+R	⌘+R
CD de recherche	Ctrl+K	⌘+K
Consulter le site	Ctrl+L	⌘+L
Scanner	Ctrl+Y	⌘+Y
Cluster	Ctrl+U	⌘+U
Recherche dans le navigateur	Ctrl+Shift+L	⌘+↑+L
Rechercher des pistes similaires	Ctrl+T	⌘+T
Générer des empreintes digitales AcoustID	Ctrl+Shift+Y	⌘+↑+Y
Balises à partir des noms de fichiers	Ctrl+Shift+T	⌘+↑+T

Aide

Action	Windows / Linux	macOS
Aide	F1	⌘+?
Afficher l'historique des activités	Ctrl+H	⌘+⬆+H
Afficher le journal d'erreur/débogage	Ctrl+G	⌘+G

Vue des métadonnées

Action	Windows / Linux	macOS
Ajouter un nouveau tag	Alt+Shift+A	⌥+⬆+A
Modifier la balise sélectionnée	Alt+Shift+E	⌥+⬆+E
Supprimer la balise sélectionnée	Alt+Shift+R Del	⌥+⬆+R Del ⌘+⌫
Copier la valeur de la balise sélectionnée	Ctrl+C	⌘+C
Coller à la valeur de la balise sélectionnée	Ctrl+V	⌘+V

Autre

Action	Windows / Linux	macOS
Recherche ciblée	Ctrl+F	⌘+F
Supprimer l'élément sélectionné	Del	Del ⌘+⌫

17.4.2 Éditeur de script

Action	Windows / Linux	macOS
Afficher la complétion automatique	Ctrl+Space	^+Space
Utiliser l'achèvement sélectionné	Tab Return	Tab Return
Cacher les achèvements	Esc	Esc

Symboles

-M command line option, 247
-N command line option, 247
-P command line option, 247
-V command line option, 247
--config-file CONFIG_FILE command line option, 246
--debug command line option, 247
--help command line option, 246
--long-version command line option, 247
--no-player command line option, 247
--no-plugins command line option, 247
--no-restore command line option, 247
--version command line option, 247
-c CONFIG_FILE command line option, 246
-d command line option, 247
-h command line option, 246
-v command line option, 247

A

AcoustID, 7 soumission, 162
acoustid utilisation, 195
actions du menu contextuel plugins, 222
album erreur de chargement, 186
albumartist, 7 analyser les fichiers, 138
android app, 204
api plugins, 217
archives d'art de couverture, 8
archives de l'art de la couverture configuration, 45
art de couverture configuration, 42

B
balisage scripts, 171
balises avancées, 78
 de base, 73
 de genre, 79
 de musique classique, 83
 des plugins, 84
 mappage, 223
 écriture, 87
balises à partir des noms de fichiers, 166
barre d'état, 22

C

CAA, see archives d'art de couverture
chercher cd, 133
command line option
-M, 247
-N, 247
-P, 247
-V, 247
--config-file CONFIG_FILE, 246
--debug, 247
--help, 246
--long-version, 247
--no-player, 247
--no-plugins, 247
--no-restore, 247
--version, 247
-c CONFIG_FILE, 246
-d, 247
-h, 246
-v, 247
configuration
archives de l'art de la couverture, 45
art de couverture, 42
avant le balisage, 34
configuration de l'écran, 23
couleurs, 61
dénomination des fichiers, 48
emplacement de la pochette, 42
emplacement du fichier de configuration, 190
empreinte digitale, 51
fichier, 190
fichiers locaux, 47
fournisseurs d'art de couverture, 44
genres, 30
interface utilisateur, 59
options avancées, 66
options d'action, 24
options de balise, 33
options de balise aac, 37
options de balise ac3, 39
options de balise id3, 35
options de balise wave, 40
options de métadonnées, 27
options générales, 25

plugins, 55
préférences correspondantes, 70
préférences de version, 29
recherche cd, 52
réseau, 68
scripts, 64
top tags, 63
vérification des mises à jour, 26
évaluations, 32
contribuer, 4
crédits d'artiste, 8

D

dénomination des fichiers
configuration, 48
dépannage
aucune pochette affichée, 181
aucune pochette téléchargée, 180
balises non enregistrées, 181
fichiers non enregistrés, 182
général, 177
journaux, 178
le programme ne démarre pas, 178
le programme se fige, 183
macOS montre que l'application est endommagée, 183
signaler un bug, 177

E

empreinte acoustique, 7, 190
empreinte digitale
acoustique, 190
soumission, 162
empreinte digitale acoustique
soumission, 155, 156, 162
enregistrement, 9
enregistrement de fichiers, 151
erreur
impossible de charger l'album, 186

F

fichiers
enregistrement, 151
fichiers de recherche, 136
flatpak
installation, 12

flux de travail
 fichiers non groupés, 156
fonctions de script, 91
 affectation, 91
 boucle, 130
 conditionnel, 118
 divers, 131
 informations, 129
 mathématique, 115
 multi-valeur, 107
 plugins, 221
 texte, 95
format de fichier, 187
 plugins, 220

G

glossaire, 7

H

hooks d'événement
 plugins, 219

I

icône
 tagger, 185
icône du tagueur, 185
icônes
 album, 20
 barre d'état, 22
 libération, 20
icônes d'état, 20
icônes de piste, 21
id de disque
 attacher, 159
id du disque, 8
install
 Paquet Linux, 12
installation
 flatpak, 12
 snap, 12
 télécharger, 11
interface utilisateur
 configuration, 59
 couleurs, 61
 raccourcis clavier, 248
 écran principal, 14
itunes, 191
 étiquettes, 191

L

l'artiste, 8
L'identifiant MusicBrainz, 9
lecteur audio, 189
ligne de commande
 options, 246
limites, 3

M

macos
 app endommagée, 187
 dossiers réseau, 186
mappage
 balises, 223

marquage
 itunes, 191
mbid, see L'identifiant MusicBrainz
mise en correspondance des
 fichiers aux pistes, 147

mots clés
 édition, 189

moyen, 9

N

nom de fichier
 scripts, 171
non-album track, see standalone
 recording

O

options
 ligne de commande, 246
ordre de traitement, 175

P

paire: acoustique
 empreinte digitale, 195
paire: configuration
 navigateur, 191
Paquet Linux
 install, 12
paramètres des options,
 configuration
pas de métadonnées, 157
pays de sortie
 plusieurs, 197
plugin
 écriture, 199

plugins, 169
 actions du menu contextuel, 222
 api, 217
 configuration, 55
 fonctions de script, 221
 format de fichier, 220
 hooks d'événement, 219
 installation, 58
 métadonnées, 217
 processeurs de métadonnées, 218
 programmation, 217
 tierce personne, 55
 types, 169
plusieurs pays de sortie, 197
pochette
 emplacement pour enregistrer, 42
 réglage, 149
programmation
 plugins, 217

R

recherche dans le navigateur, 140
recording
 standalone, 10
release, 9
 supprimé, 186
release group, 9
remerciements, 5

S

script de nommage de fichier, 192
scripts, 89, 171
 balisage, 64, 171
 balises, 72
 dénomination de fichier, 50
 nom de fichier, 171
 syntaxe, 89
 variables, 72
snap
 installation, 12
standalone recording, 10

T

tags
 tronqué, 191
track, 10

V

variables

avancées, 83
de base, 80
de fichier, 82
version
 erreur de chargement, 186
vérification des mises à jour
 configuration, 26

W

WMP
 balises, 191
work, 10
workflows
 cd, 154
 fichiers regroupés par album,
 155
 général, 153