

---

# MusicBrainz Picard

*Version v2.3.2*

nov. 21, 2020

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>

---

## Table des matières

---

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Crédits et mises à jour de la documentation . . . . .	2
1.2	Picard can . . . . .	2
1.3	Picard ne peut pas . . . . .	3
1.4	Limites . . . . .	3
<b>2</b>	<b>Glossaire des termes</b>	<b>4</b>
<b>3</b>	<b>Commencer</b>	<b>7</b>
3.1	Téléchargez et installez Picard . . . . .	7
3.2	Écran principal . . . . .	8
3.3	Icônes d'état . . . . .	9
<b>4</b>	<b>Configuration</b>	<b>11</b>
4.1	Configuration d'écran . . . . .	11
4.2	Options d'action . . . . .	12
4.3	Paramètres des options . . . . .	12
<b>5</b>	<b>Balises et variables</b>	<b>51</b>
5.1	Balises de base . . . . .	52
5.2	Balises avancées . . . . .	56
5.3	Variables de base . . . . .	57
5.4	Variables avancées . . . . .	59
5.5	Tags de musique classique . . . . .	59
5.6	Balises des plugins . . . . .	60
5.7	Les autres informations . . . . .	63
<b>6</b>	<b>Scripting</b>	<b>64</b>
6.1	Syntaxe . . . . .	64
6.2	Variables de métadonnées . . . . .	65
<b>7</b>	<b>Fonctions de script</b>	<b>66</b>
7.1	Fonctions d'affectation . . . . .	66

7.2	Fonctions de texte . . . . .	69
7.3	Fonctions à valeurs multiples . . . . .	81
7.4	Fonctions mathématiques . . . . .	86
7.5	Fonctions conditionnelles . . . . .	89
7.6	Fonctions d'information . . . . .	100
7.7	Fonctions de boucle . . . . .	101
7.8	Fonctions diverses . . . . .	102
<b>8</b>	<b>Utiliser Picard</b>	<b>103</b>
8.1	Récupération des informations sur l'album . . . . .	103
8.2	Faire correspondre les fichiers aux pistes . . . . .	116
8.3	Définition de la pochette . . . . .	118
8.4	Enregistrement des fichiers mis à jour . . . . .	119
<b>9</b>	<b>Recommandations de flux de travail</b>	<b>121</b>
9.1	Lorsque le CD est disponible . . . . .	121
9.2	Lorsque les fichiers sont regroupés par album . . . . .	122
9.3	Lorsque les fichiers ne sont pas groupés mais ont des métadonnées . . . . .	123
9.4	Lorsque les fichiers ne sont pas regroupés et ont peu ou pas de métadonnées existantes . . . . .	124
<b>10</b>	<b>Autres tâches de Picard</b>	<b>126</b>
10.1	Joindre un ID de disque à une version . . . . .	126
10.2	Soumission d'empreintes acoustiques . . . . .	128
<b>11</b>	<b>Prolonger Picard</b>	<b>133</b>
11.1	Plugins . . . . .	133
11.2	Les scripts . . . . .	134
11.3	Commande en traitement . . . . .	135
<b>12</b>	<b>Dépannage</b>	<b>138</b>
12.1	Dépannage général . . . . .	138
12.2	There is no coverart . . . . .	139
12.3	Les balises ne sont ni mises à jour ni enregistrées . . . . .	140
12.4	Les fichiers ne sont pas enregistrés . . . . .	141
12.5	Picard vient d'arrêter de travailler . . . . .	141
<b>13</b>	<b>Questions fréquemment posées</b>	<b>143</b>
13.1	Utiliser Picard . . . . .	143
13.2	Formats de fichier . . . . .	144
13.3	Configuration . . . . .	146
<b>14</b>	<b>Annexes</b>	<b>148</b>
14.1	Annexe 1 : API des plugins . . . . .	148
14.2	Annexe 2 : Mappage des balises . . . . .	154
14.3	Annexe 3 : Options de ligne de commande . . . . .	177
	<b>Index</b>	<b>178</b>

# CHAPITRE 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 est destiné à compléter et éventuellement à remplacer les informations fournies sur le [site Web de Picard](#) et à les rendre également disponibles 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 contenues dans 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 pléthore de personnalisations pour adapter les collections de musique à vos besoins.* »

## 1.1 Crédits et mises à jour de la documentation

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 communautaire, 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. Les contributeurs individuels ne sont pas répertoriés ici par peur de manquer quelqu'un ; cependant, sachez que vos contributions sont grandement appréciées.

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 davantage cette documentation sera très appréciée - et 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'tirer](#) pour résoudre les problèmes en suspens sont également appréciées.

Bob Swift (rdswift)  
Éditeur

## 1.2 Picard can ...

... 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](#).

## 1.3 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.

## 1.4 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 page Web [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.

## CHAPITRE 2

---

### 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 compositeurs, 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 de 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, `1wH18fGzJyLXQR33ug60E8jh4k-` 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.

### 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. Par exemple « Be (One in a Million) » (identifiant d'enregistrement : `fbd0e1a0-b6c5-4926-b89d-3922e4144da9`) est l'enregistrement lié à la piste 14 de l'album « Tourist ». Chaque piste doit toujours être associée à un seul enregistrement, mais un enregistrement peut être lié à n'importe quel nombre de pistes.

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](#) » (ID de version : 4e4ba41e-24ae-3f57-87f6-3d8f19ae9483) est une version de l'album sorti 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. Release groups 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 release group embrasse le concept général d'un album - il peu importe le nombre de CD ou d'éditions / versions dont il dispose. Par exemple, « [Sea of No Cares](#) » (ID de version : 4e4ba41e-24ae-3f57-87f6-3d8f19ae9483) est une version de l'album dans le [groupe de publication id 7e7ffd2b-3d1b-3487-aaaf-e4e6037f09ca](#) par Great Big Sea.

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 release. Un exemple est « [Sea of No Cares \(live\)](#) » (ID d'enregistrement : 0198c132-ed38-430c-92bd-d3c7e9ff25b8) par 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](#) » (identifiant de piste : bf8ecb3c-6fe6-41b7-a078-5748265a9f94).

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. Par exemple, une œuvre peut être un roman, une pièce de théâtre, un poème ou un essai, enregistré plus tard sous forme d'oratoire ou de livre audio. « [Blinded by the Light](#) » de « [Manfred Mann's Earth Band](#) » est un enregistrement de l'[ID de travail 7a757d97-da2a-3751-8d32-94d471de2eeb](#) « [Blinded by the Light](#) » écrit par Bruce Springsteen.

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](#).

### 3.1 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](#).

### 3.1.1 Linux : installation avec Flatpak

Picard est également disponible sur Flathub. Cette version devrait fonctionner sur toutes les distributions Linux modernes, tant que Flatpak est installé.

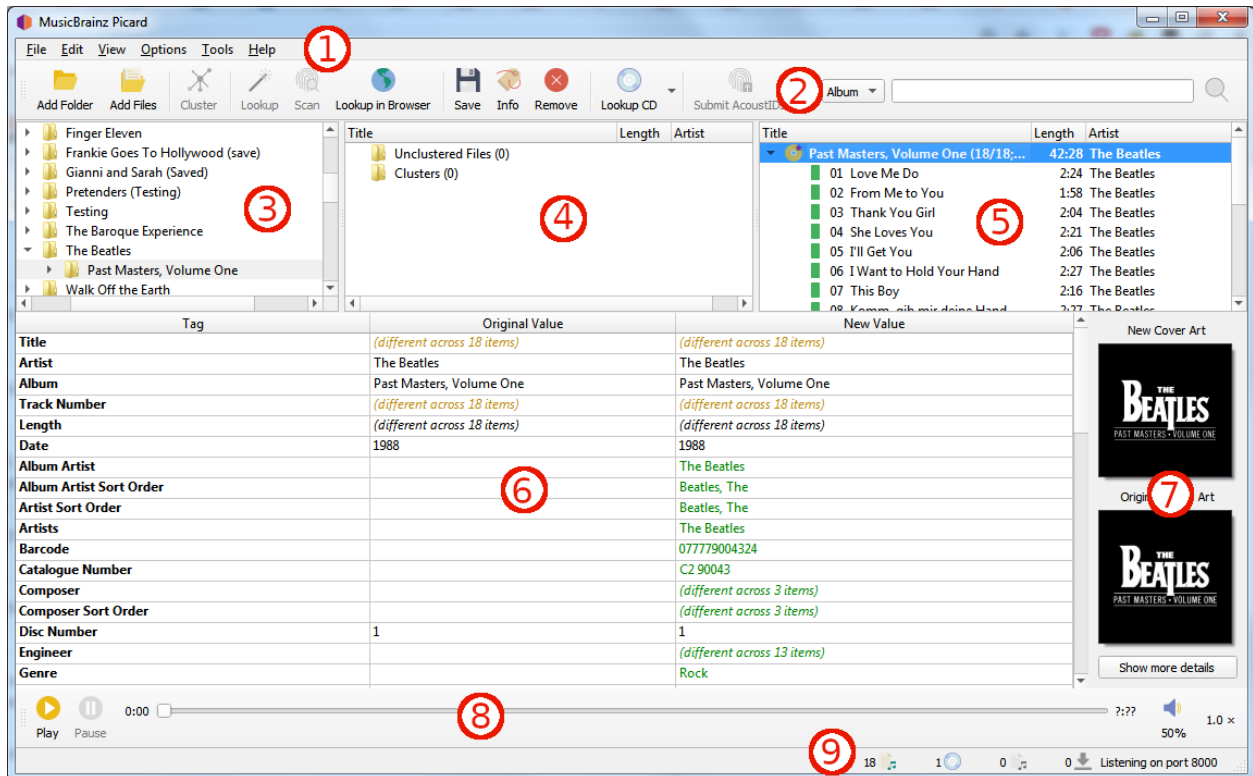
Commencez par activer le référentiel Flathub :

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

Vous pouvez maintenant installer Picard :

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

### 3.2 Écran principal



1. **Barre de menus** : Ceci fournit le menu déroulant des actions que Picard peut effectuer.
2. **Tool Bar** : This provides quick links to the main functions performed by Picard. This can be customized by the user in the *User Interface Options* settings.
3. **File Browser** : 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. **Player** : The built-in player that can be used to play selected audio files.
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.

### 3.3 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.

#### 3.3.1 Album / Release Icons



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.

### 3.3.2 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.

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.

### 4.1 Configuration d'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 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.

## 4.2 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

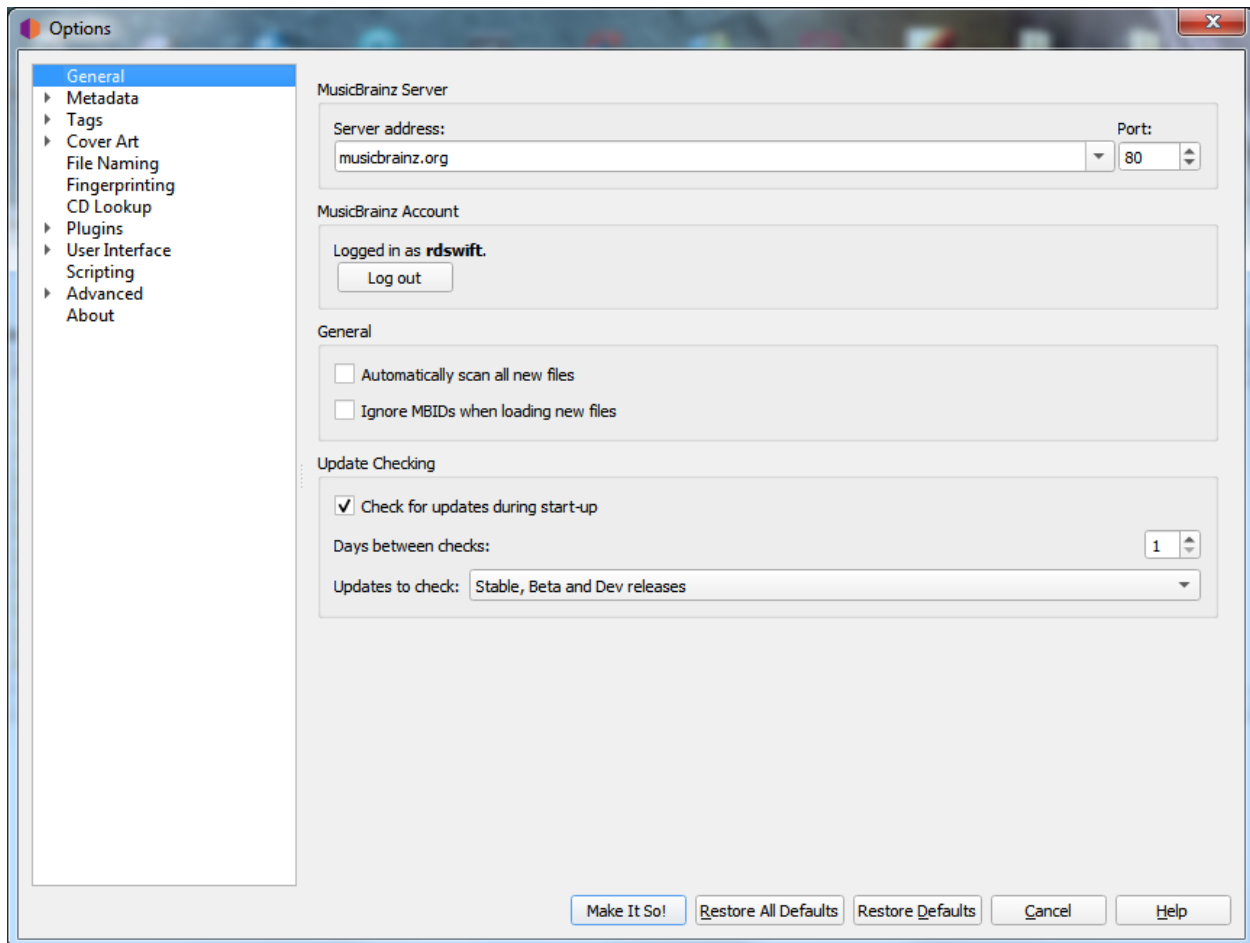
Picard mettra à jour les balises de métadonnées dans les fichiers conformément aux paramètres d'option et aux scripts de balisage spécifiés.

## 4.3 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.



### 4.3.1 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.

---

### 4.3.2 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 crédits 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.

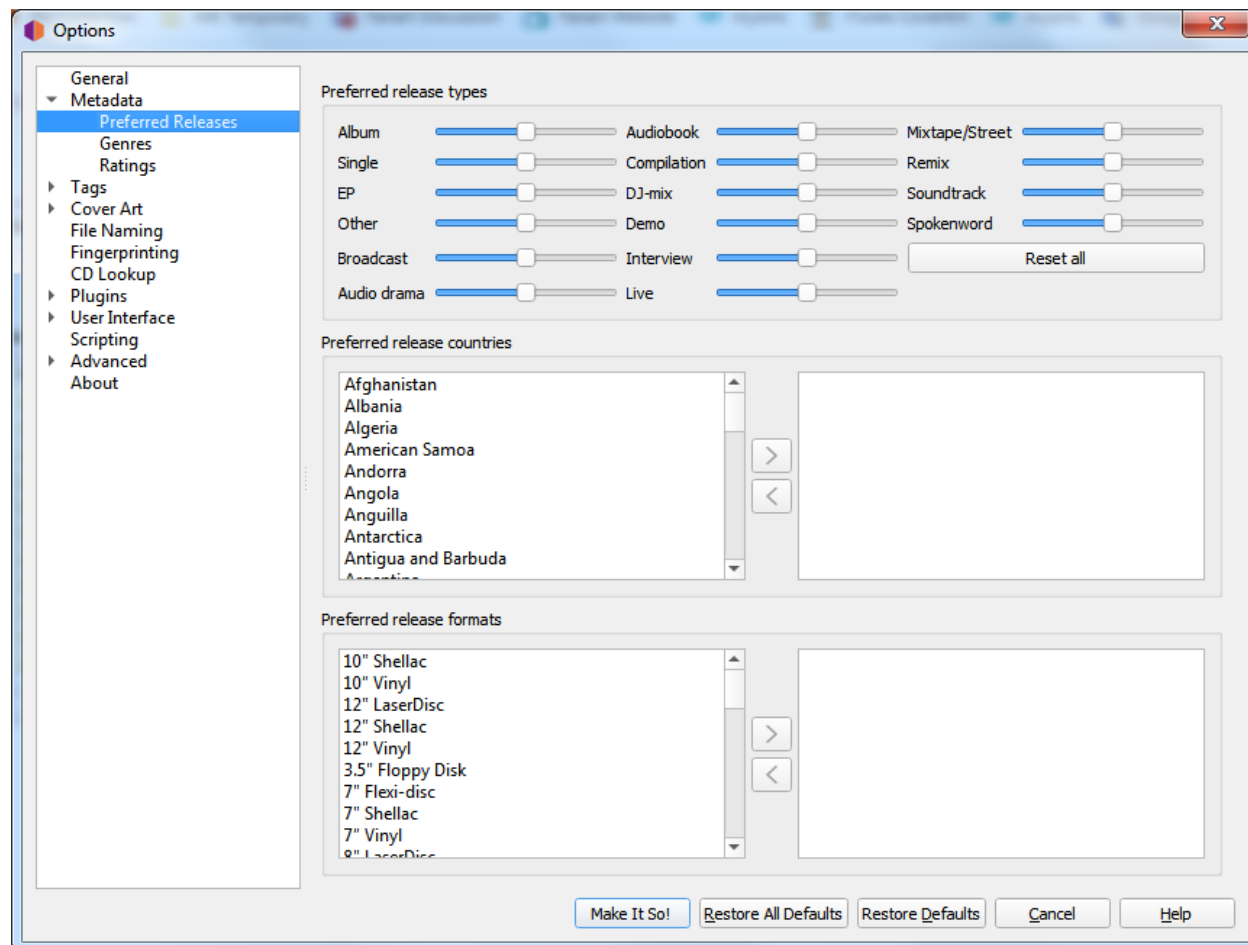
### Artistes variés

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.

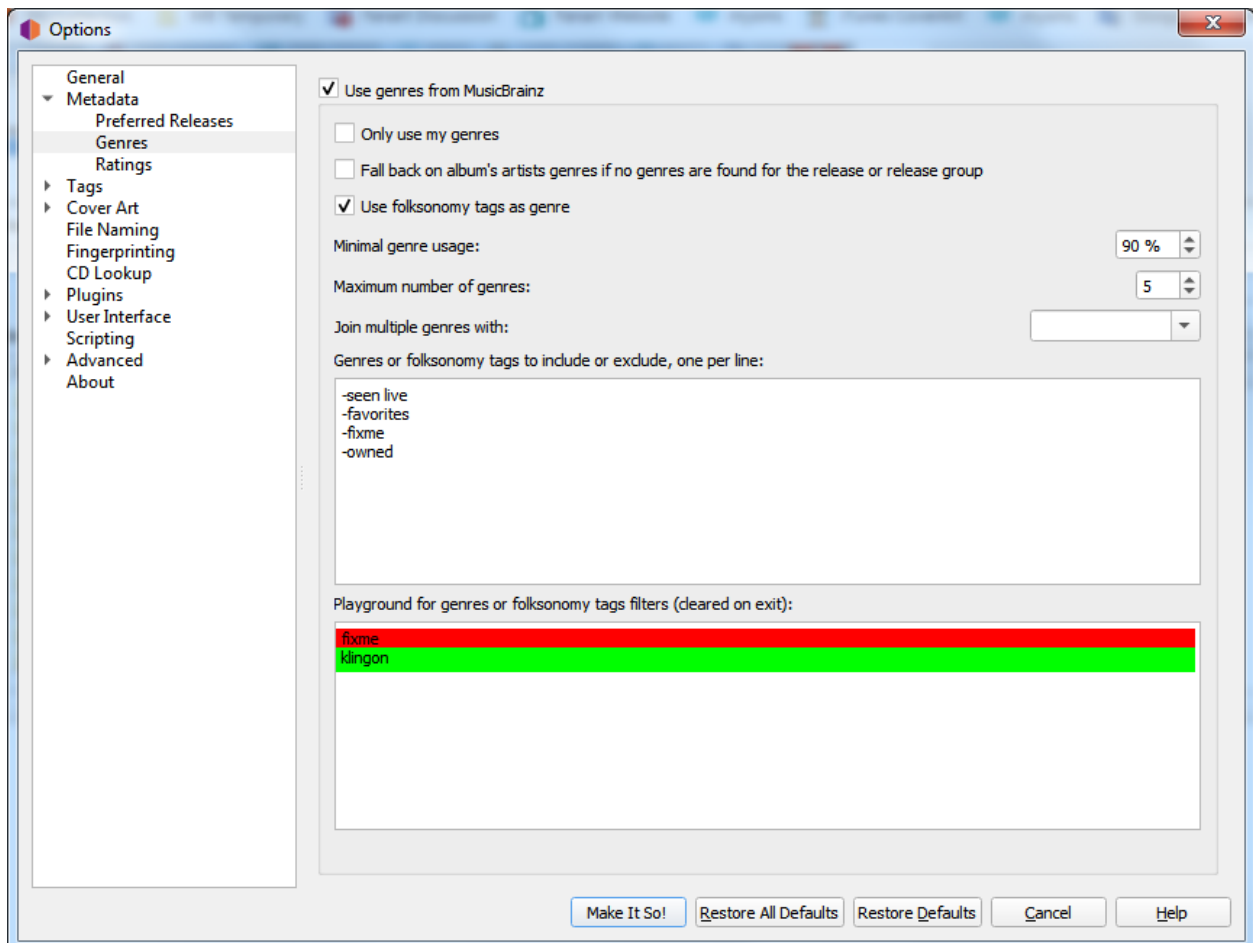
### Preferred release countries

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 ».

### Preferred release formats

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'écrit 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 :

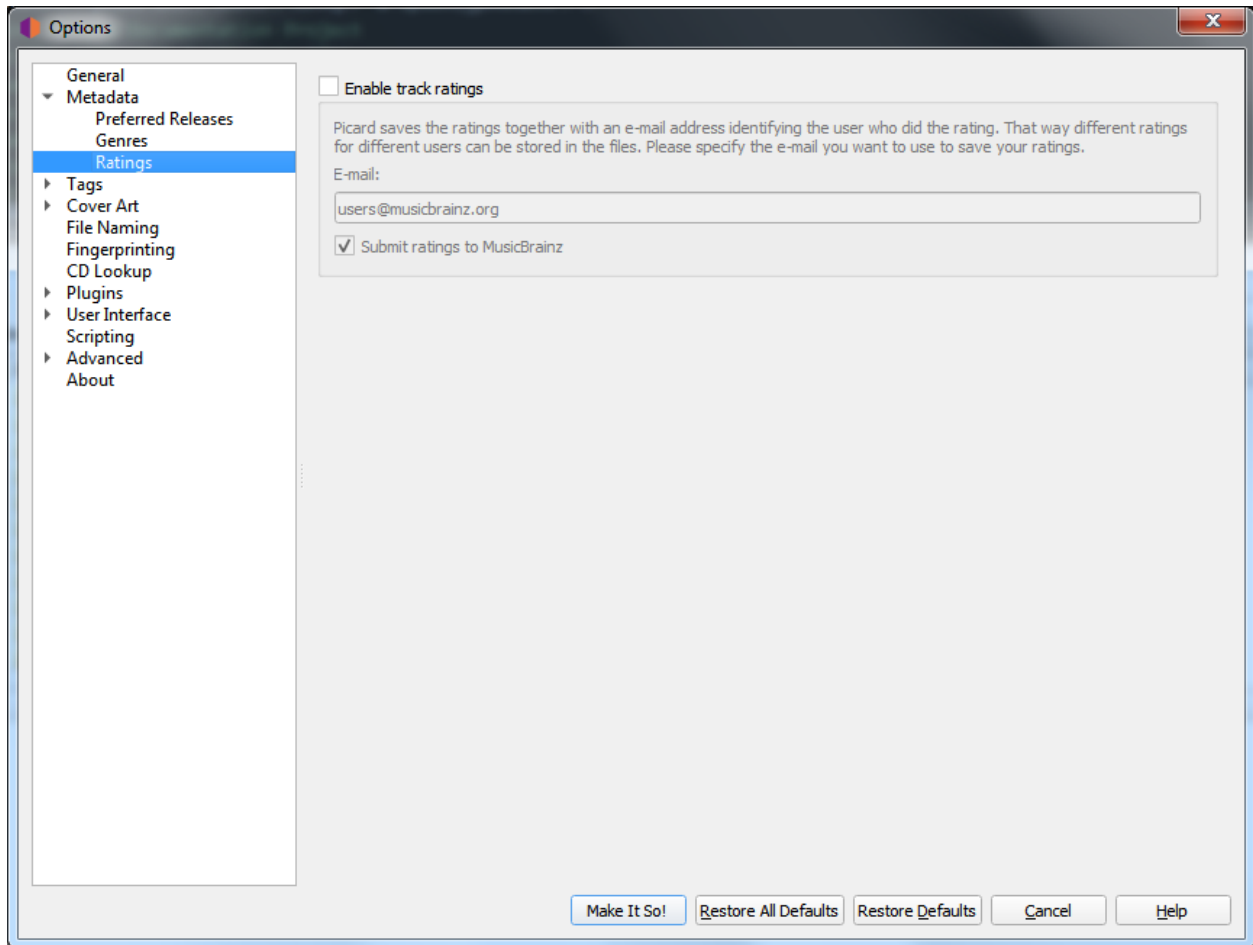
Cette zone vous permet de saisir des balises de genre, une par ligne, pour tester vos filtres. Si une étiquette est marquée en rouge, elle sera filtrée. Un tag marqué en vert sera autorisé.

---

**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.

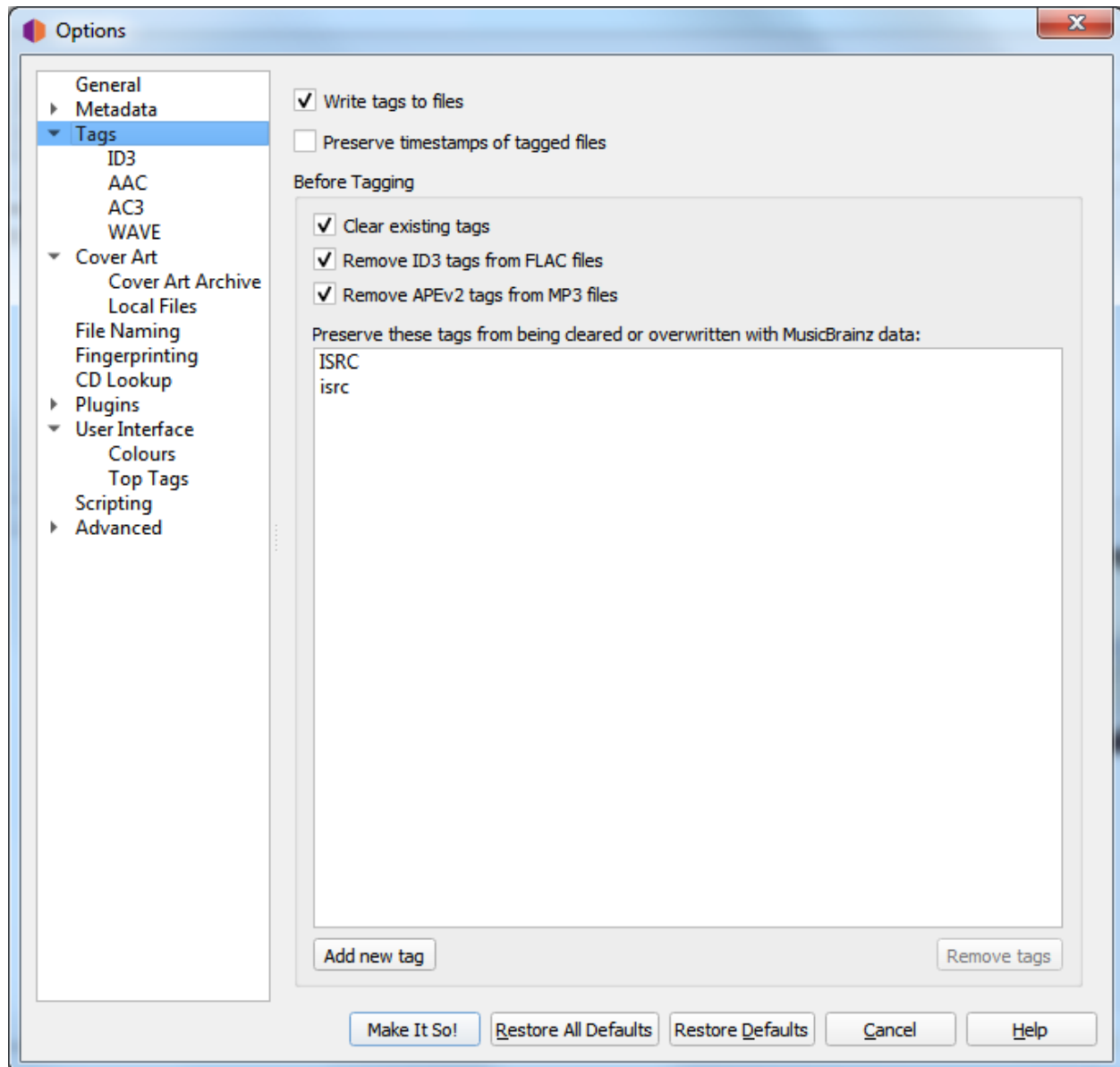
### Email

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.

### 4.3.3 Options de balise



#### Écrire des balises dans des fichiers

Uncheck to disable Picard from writing metadata to your files. Picard may still move or rename your files according to your settings.

#### 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 marquer

### 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.

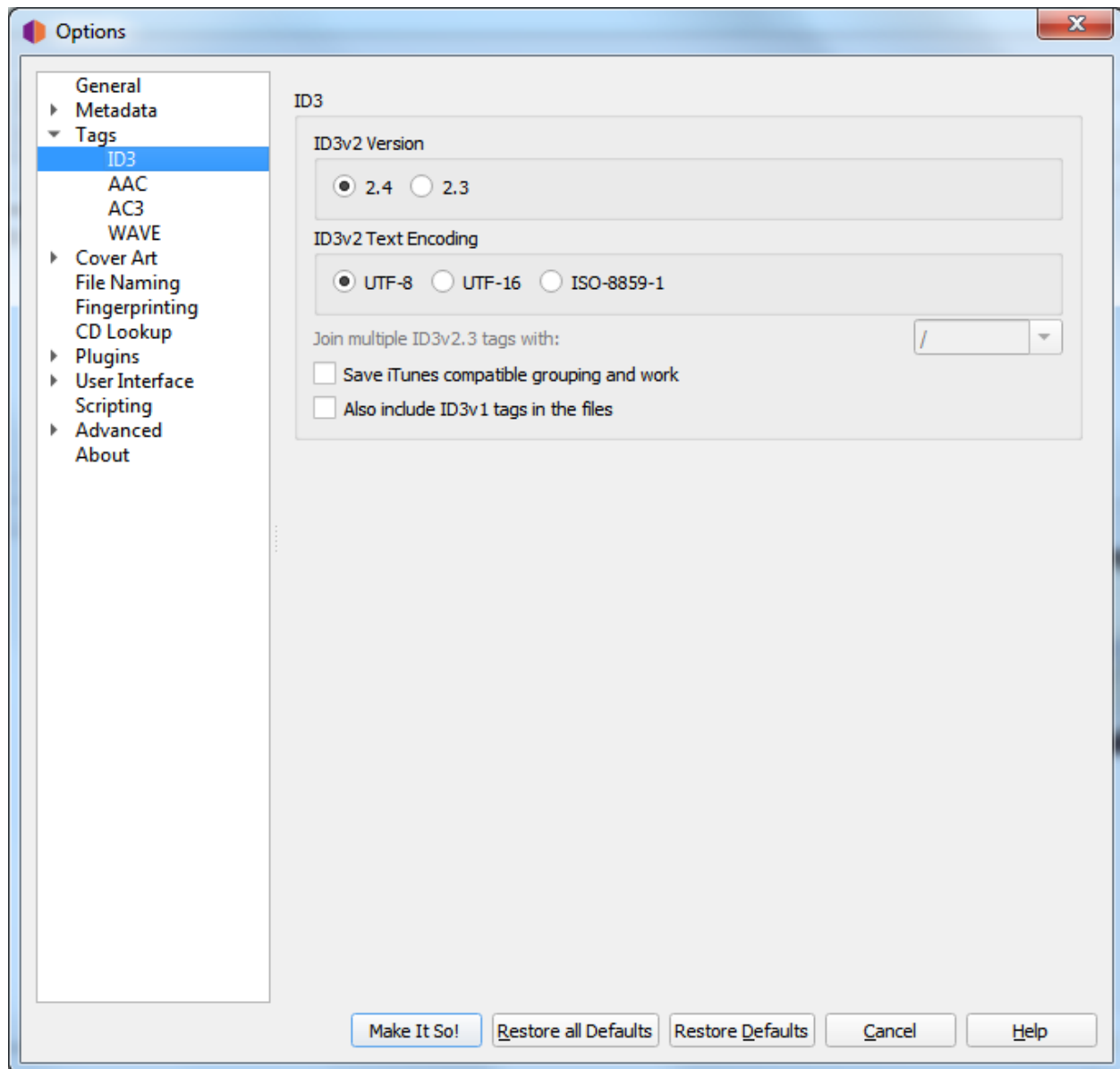
### Supprimer les balises APEv2 des fichiers MP3

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. Bien que certains logiciels n'aient aucun problème à utiliser les balises de la version 2.4, d'autres peuvent ne pas pouvoir lire les balises et 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, le [Picard Tag Mapping](#) vous montrera ce que vous perdez en choisissant la 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 aplatit 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.

### 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 page « [Picard Tag Mapping](#)” pour plus de détails.

---

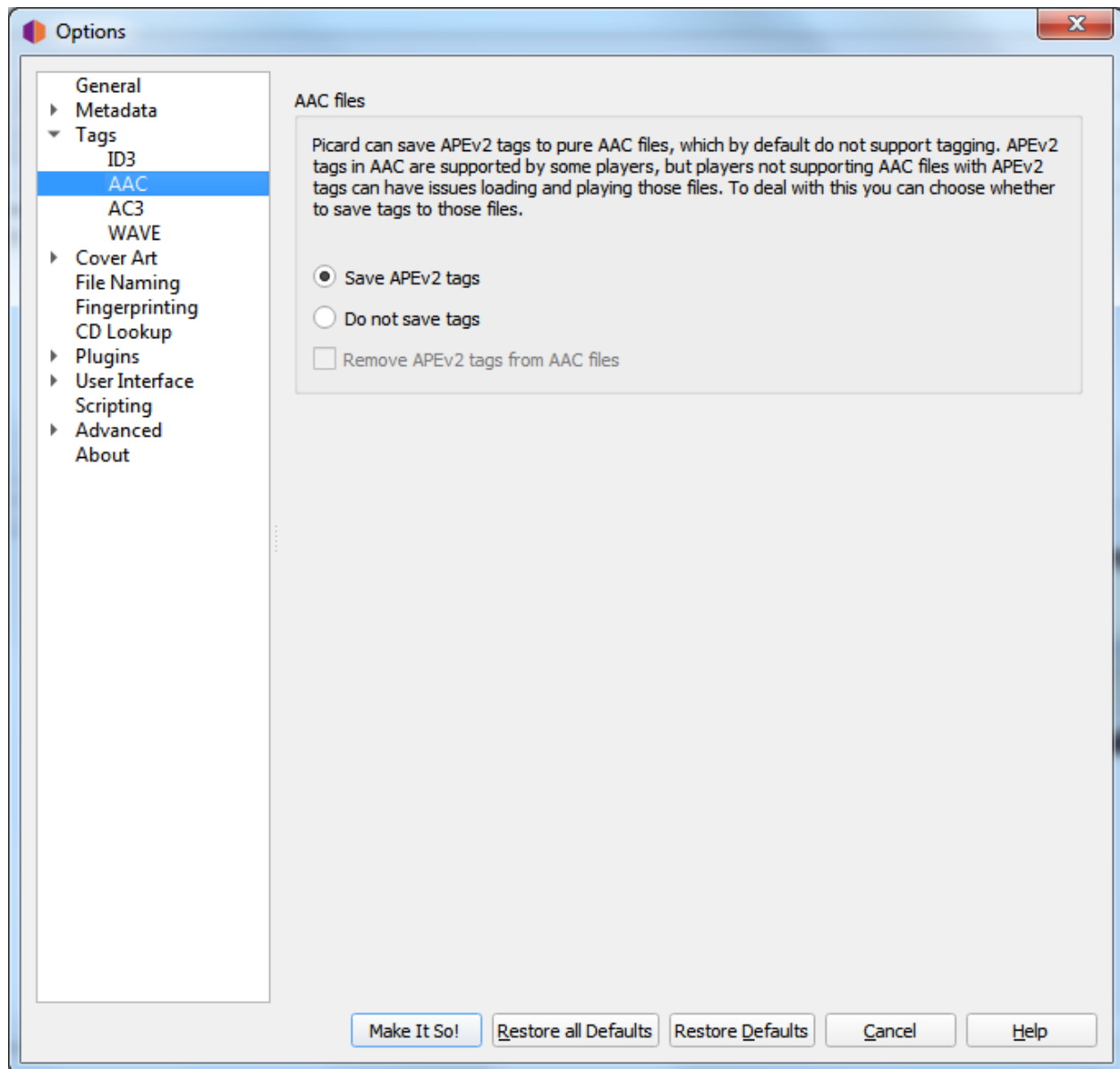
**Note :** Pour les autres joueurs prenant en charge le regroupement et le travail, vous devrez peut-être désactiver cette option. [MusicBee](#) en est un exemple.

---

**\*\* 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.

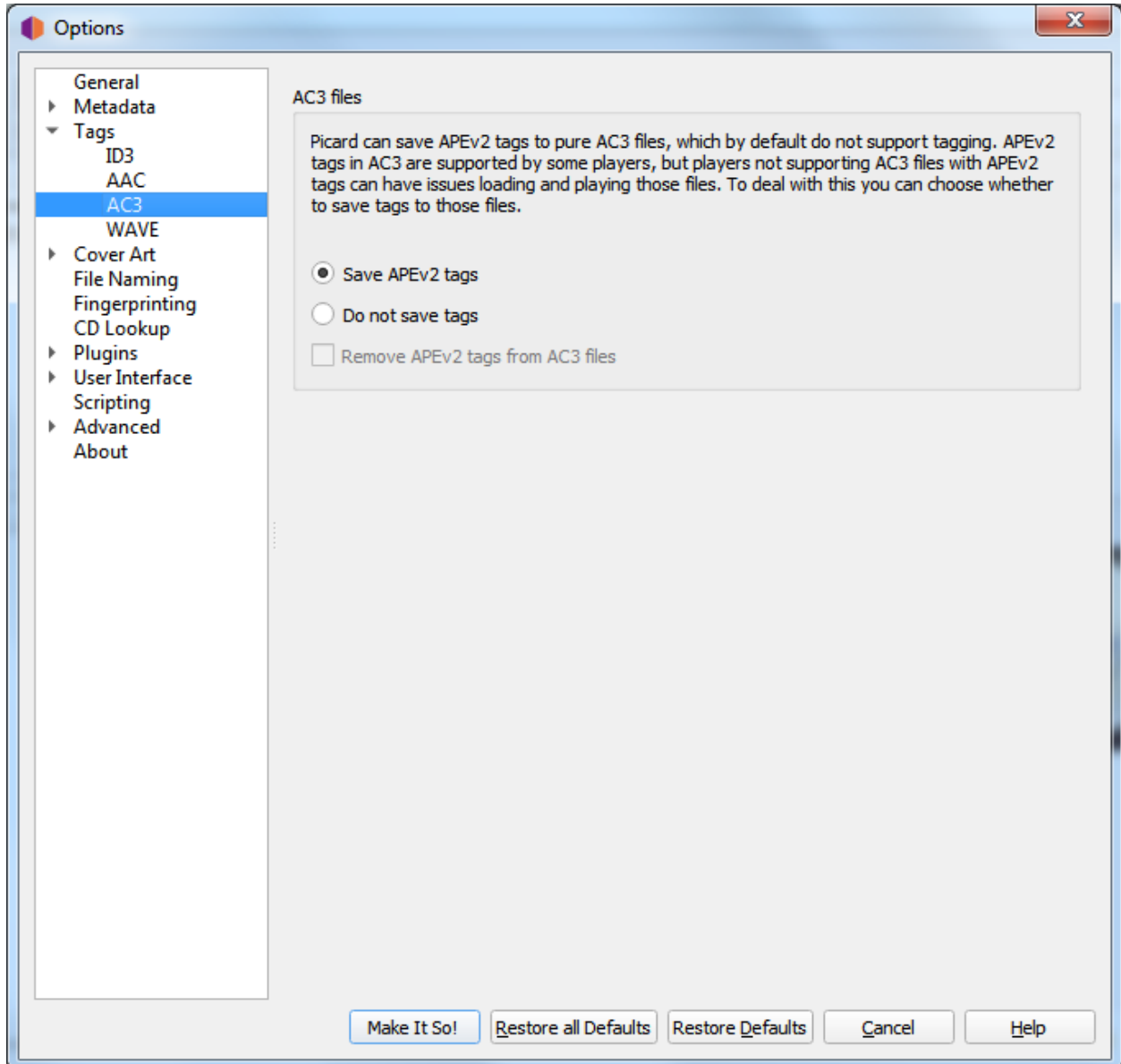
**Ne pas enregistrer les tags**

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

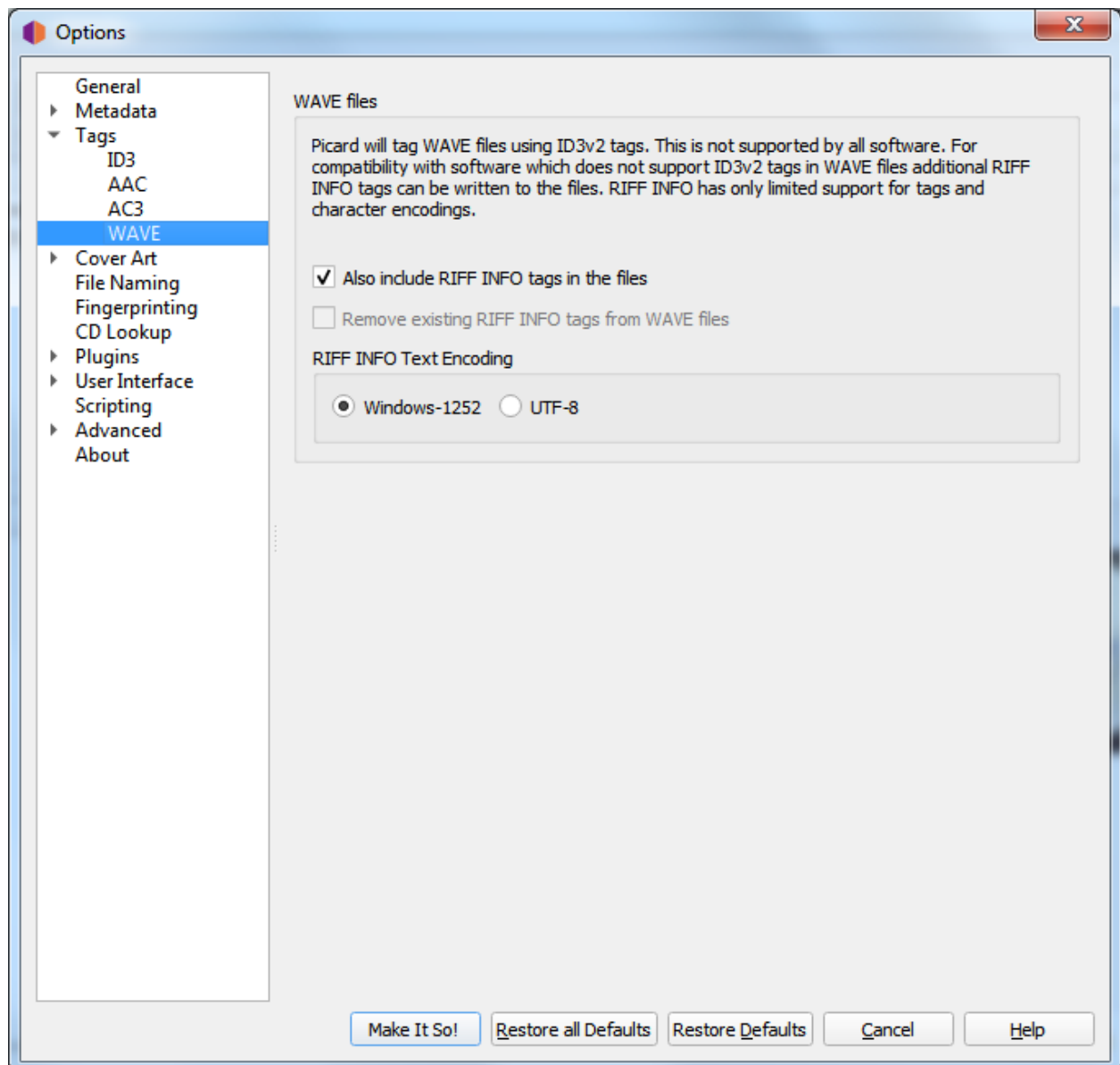
Picard n'enregistrera aucune balise dans les fichiers, mais vous pouvez toujours utiliser Picard pour les renommer. Par défaut, les balises APEv2 existantes seront conservées dans le fichier.

**\*\* 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.

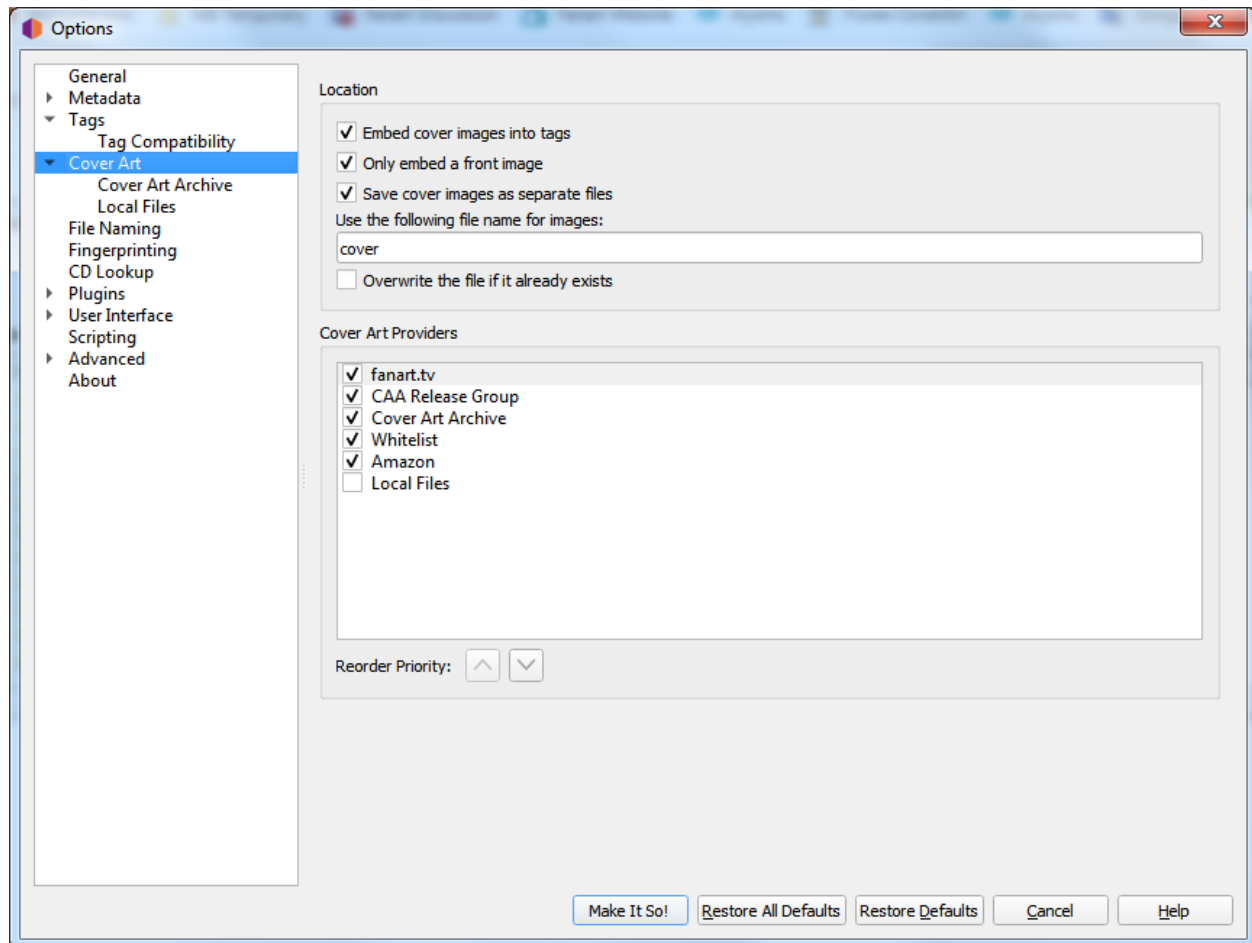
**\*\* Supprimer les balises RIFF INFO existantes des fichiers WAVE \*\***

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.

### 4.3.4 Options de couverture



---

**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.

### N'intègre qu'une image avant

N'intègre qu'une image avant dans vos fichiers musicaux. De nombreux lecteurs de musique n'afficheront qu'une seule image intégrée, donc 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 *Picard Tags* et *Picard Scripting Functions*. Le masque ne doit pas contenir d'extension de fichier ; ceci est ajouté automatiquement en fonction du type d'image réel. La valeur par défaut est « cover ». Si vous changez ceci en « cover », Windows l'utilisera pour prévisualiser le répertoire contenant.

En plus des variables de script déjà disponibles pour une piste, vous pouvez utiliser les variables spécifiques à la pochette suivantes :

- `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.

### 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.

## 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.

### Sites sur la liste blanche

Cela utilisera des images fournies par des sites sur liste blanche. Voir [Liste blanche des couvertures](#) dans le Guide de style pour plus d'informations.

---

**Note :** CD Baby et d'autres sites de liste blanche ne sont plus utilisés par MusicBrainz pour la nouvelle pochette.

---

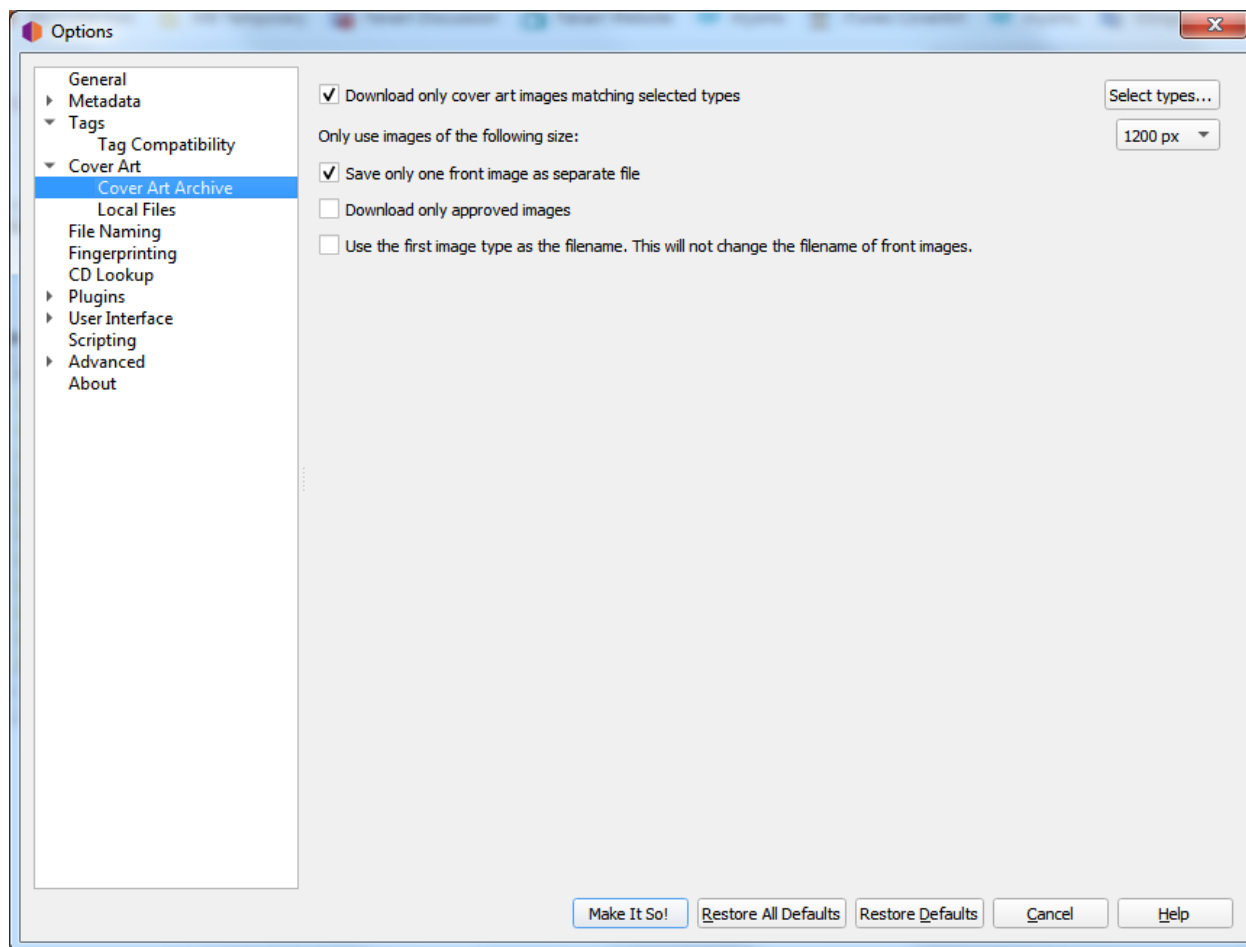
### Fichiers locaux

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](http://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](http://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.

### Archives d'art de couverture



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.

#### Types de téléchargement

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.

### **Taille de l'image**

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.

### **Enregistrez une seule image avant**

Cela indique à Picard de ne sauvegarder que la première image « avant » dans un fichier séparé avec la version. Si cette case n'est pas cochée, toutes les images « avant » seront enregistrées en tant que fichiers séparés.

### **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.

### **Utilisez le premier type d'image comme nom de fichier**

Lorsqu'elle est cochée, Picard utilisera le type de la première image récupérée comme nom de fichier lors de l'enregistrement de toutes les images. Si cette case n'est pas cochée, chaque fichier sera nommé en fonction de son type d'image.

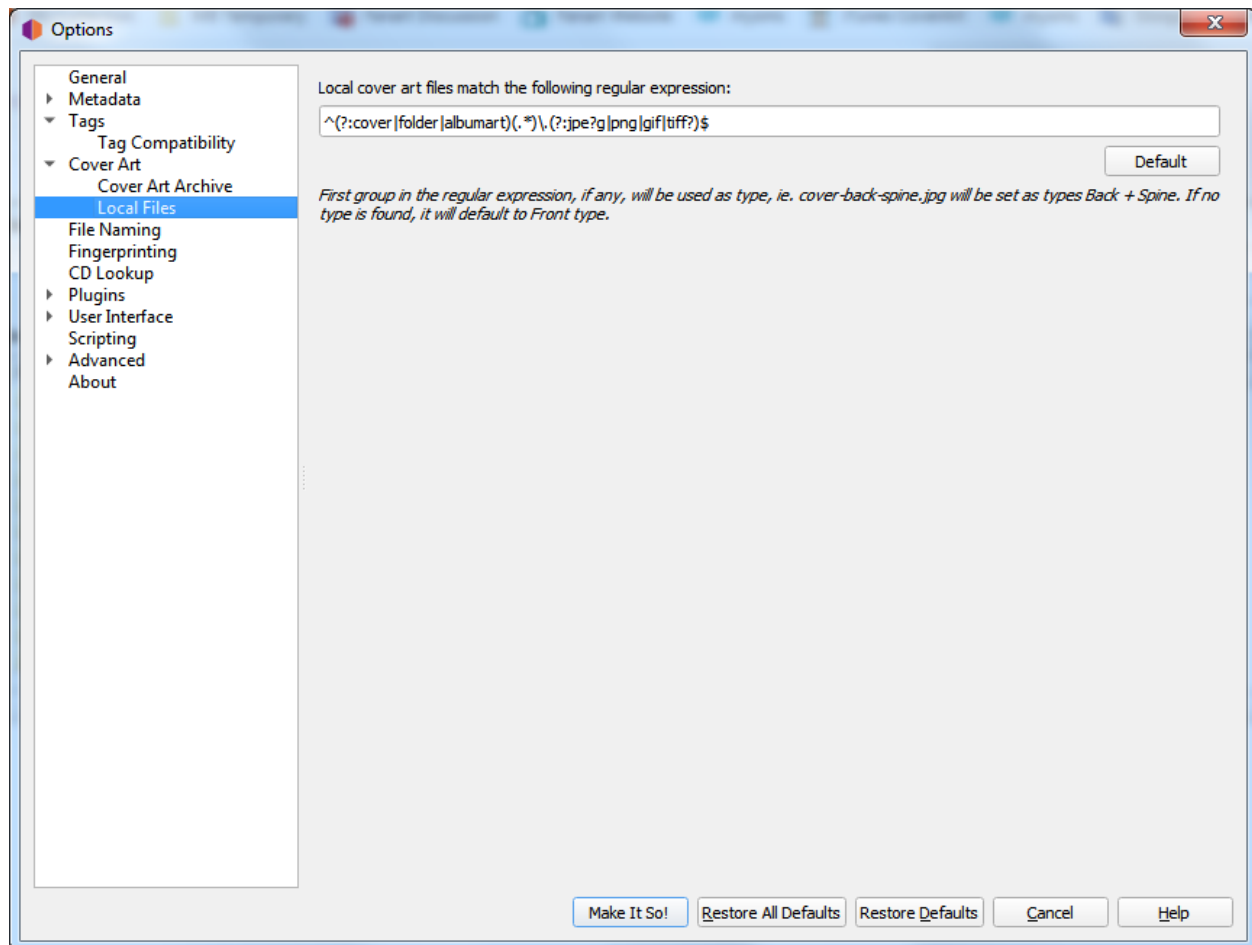
---

**Note :** Cela ne changera pas le nom utilisé pour les images « avant » qui a été spécifié dans la section *Enregistrer les images de couverture* de la section générale « *Options de la pochette* ».

---

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



In this section you can configure the file names to be used by the Local Files cover art provider. If you are trying to collect more than one image, the naming is important.

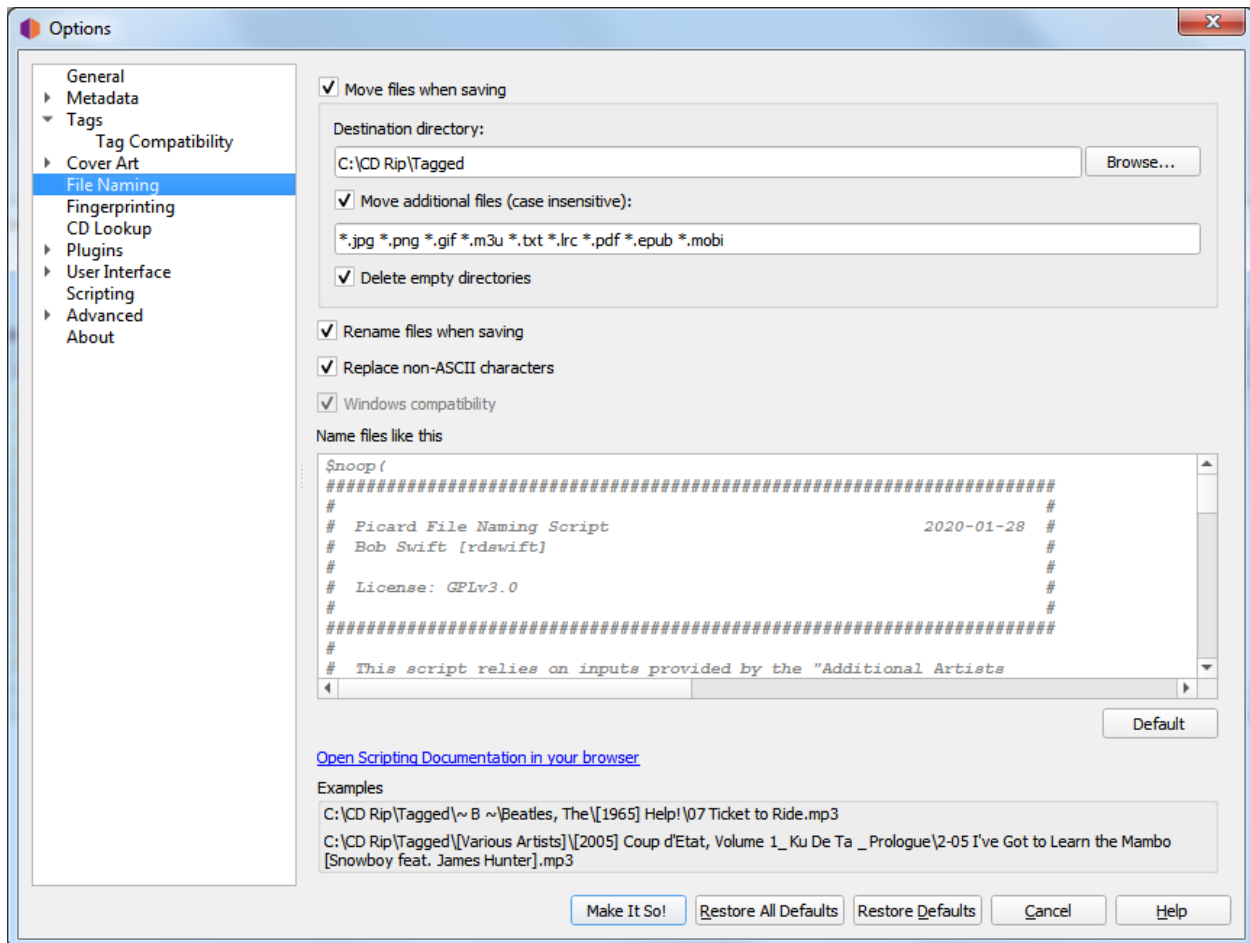
Les noms de fichiers sont définis à l'aide d'une expression régulière. La valeur par défaut est `^(?:cover|folder|albumart)(.*)\.(?:jpe?g|png|gif|tiff?)$` qui chargera les fichiers avec le nom « cover », « folder » ou « albumart » et l'extension de fichier « jpg », « png », « gif » ou « tiff » (par exemple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

### 4.3.5 Options de dénomination 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 retrouveront 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 « å » par « a » ; « é », « ë » et « ð » 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

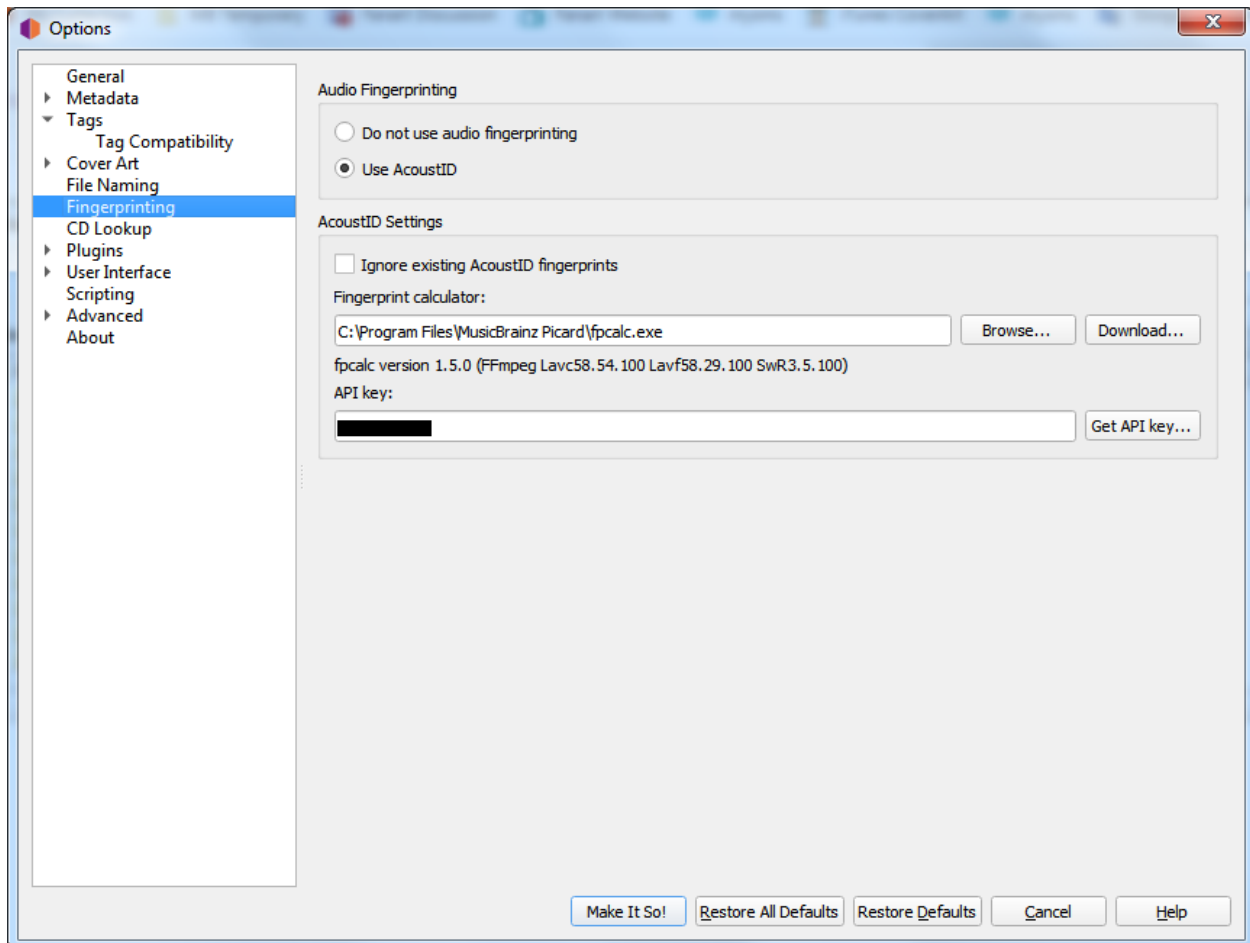
Une zone d'édition qui contient une chaîne de mise en forme qui indique à Picard quel doit être le nouveau nom du fichier et ses répertoires contenant en termes de valeurs de métadonnées diverses. La chaîne de formatage est dans *langage de script de Picard* où le texte bleu foncé commençant par un « \$ » 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.

### 4.3.6 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 digitales 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.

### **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.

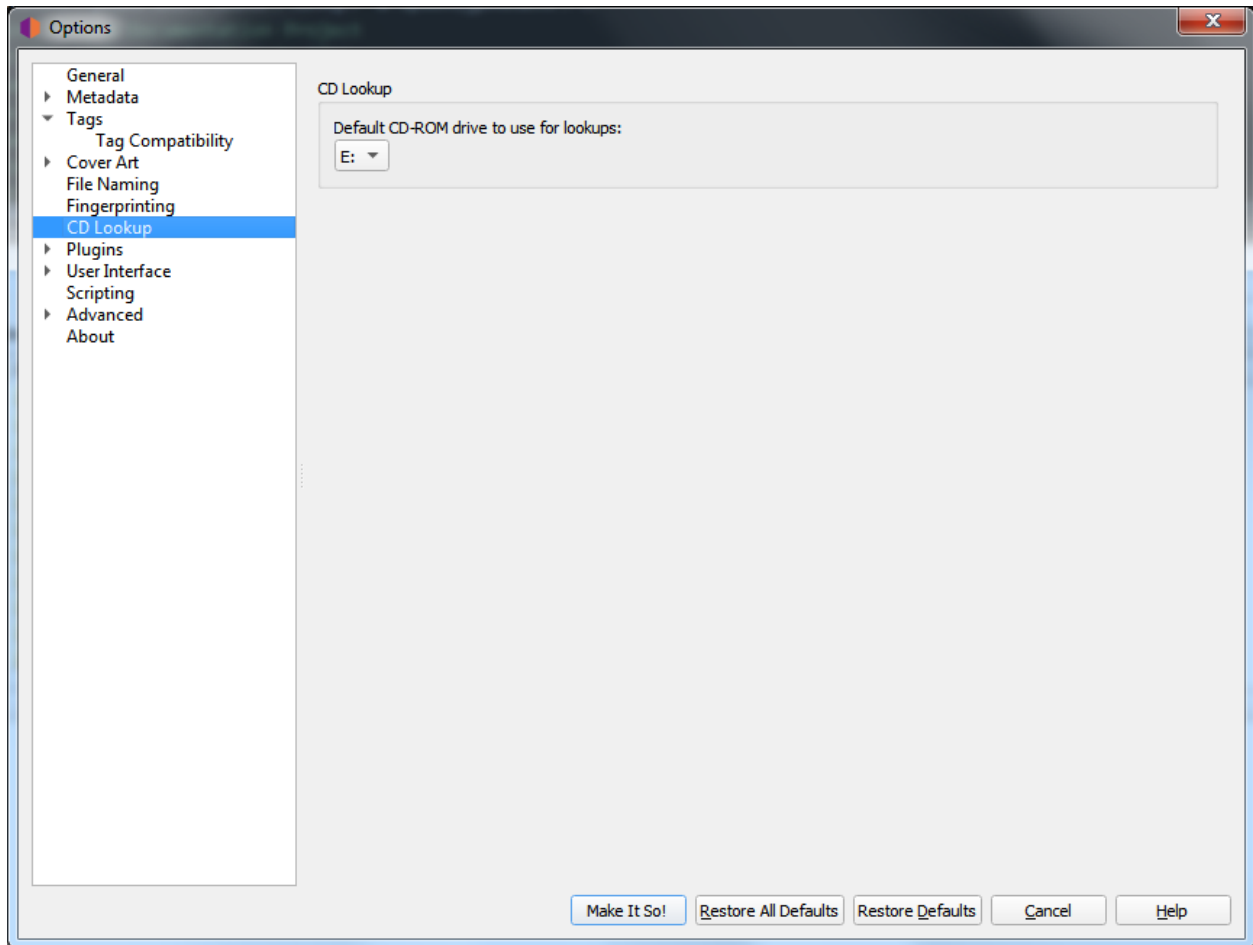
## **4.3.7 Options de recherche de CD**

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 :menuélection : "Outils --> Rechercher un CD . . ." et en sélectionnant le périphérique souhaité dans la liste des périphériques disponibles.

## macOS

Dans macOS, cette option 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

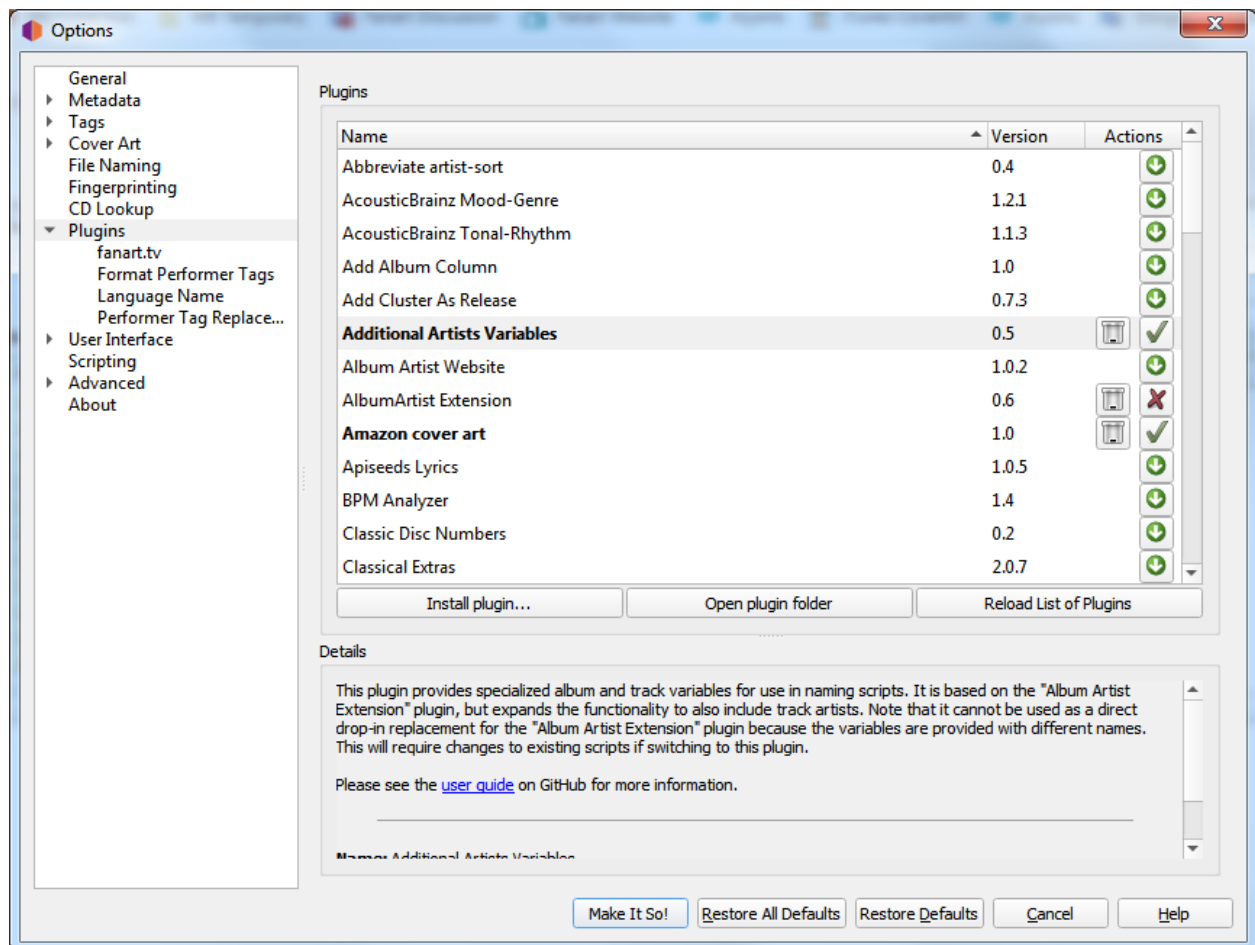
Sous Linux, Picard a un menu déroulant comme dans Windows. Si vous utilisez une ancienne version de Picard avec un champ de texte, vous devez entrer le nom de l'appareil (généralement /dev/cdrom).

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

### Autres plateformes

On other platforms, the CD Lookup option is a text field and you should enter the path to the CD drive here.

### 4.3.8 Options des plugins



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.



This icon indicates that a newer version of the plugin is available. Clicking the icon will download and install the updated version.



This icon indicates that the plugin is installed and currently enabled. Clicking the icon will disable the plugin, but it will still be installed.

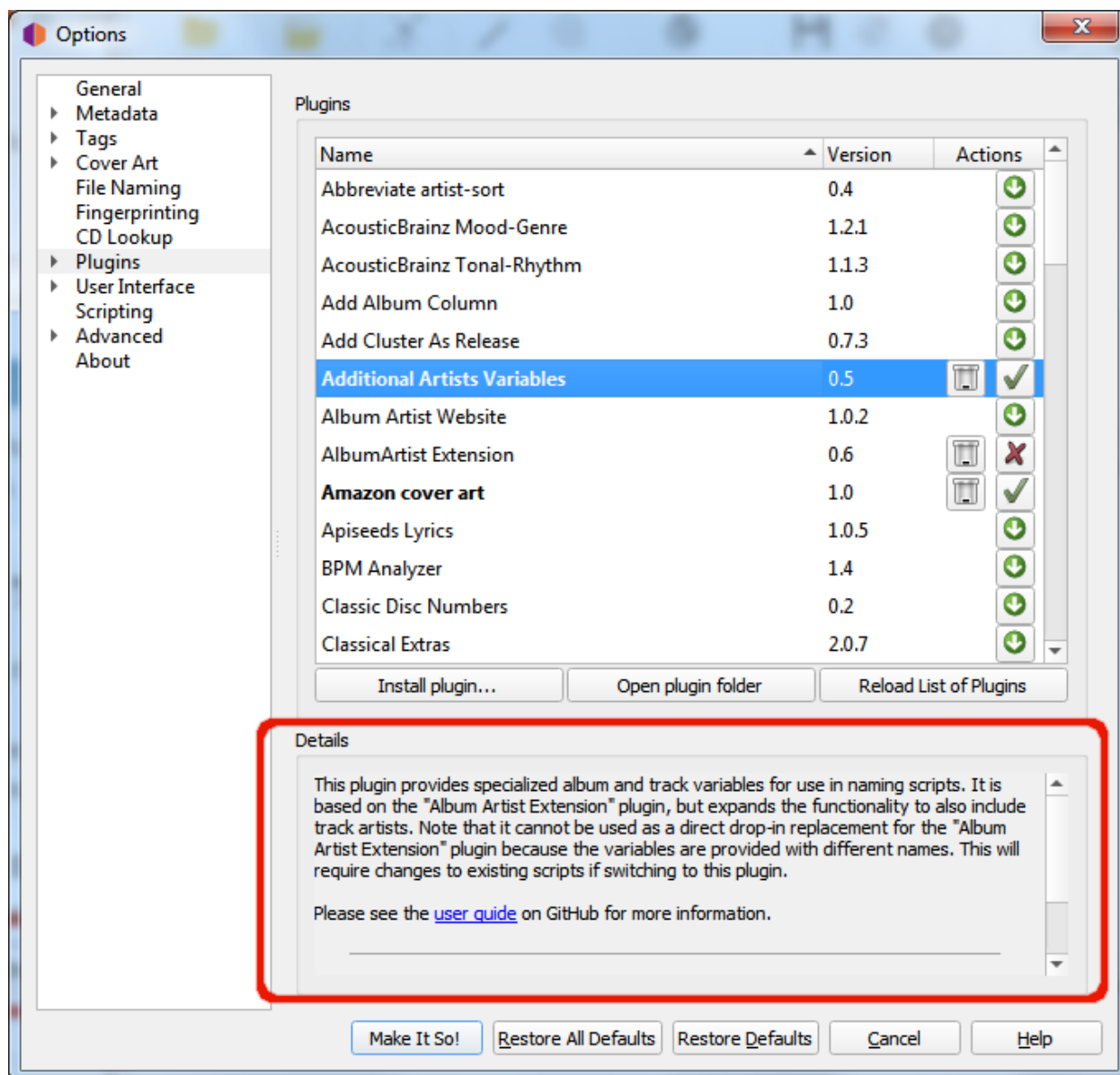


This icon indicates that the plugin is installed but currently disabled. Clicking the icon will enable the plugin.



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.

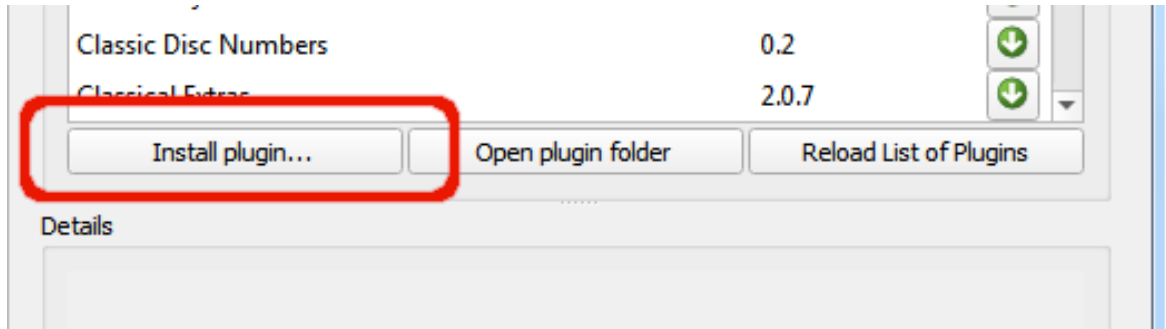


**Note :** Certains plugins ont leur propre page d'options qui apparaîtra généralement sous la section « Plugins » des options.

## Installer des plugins tiers

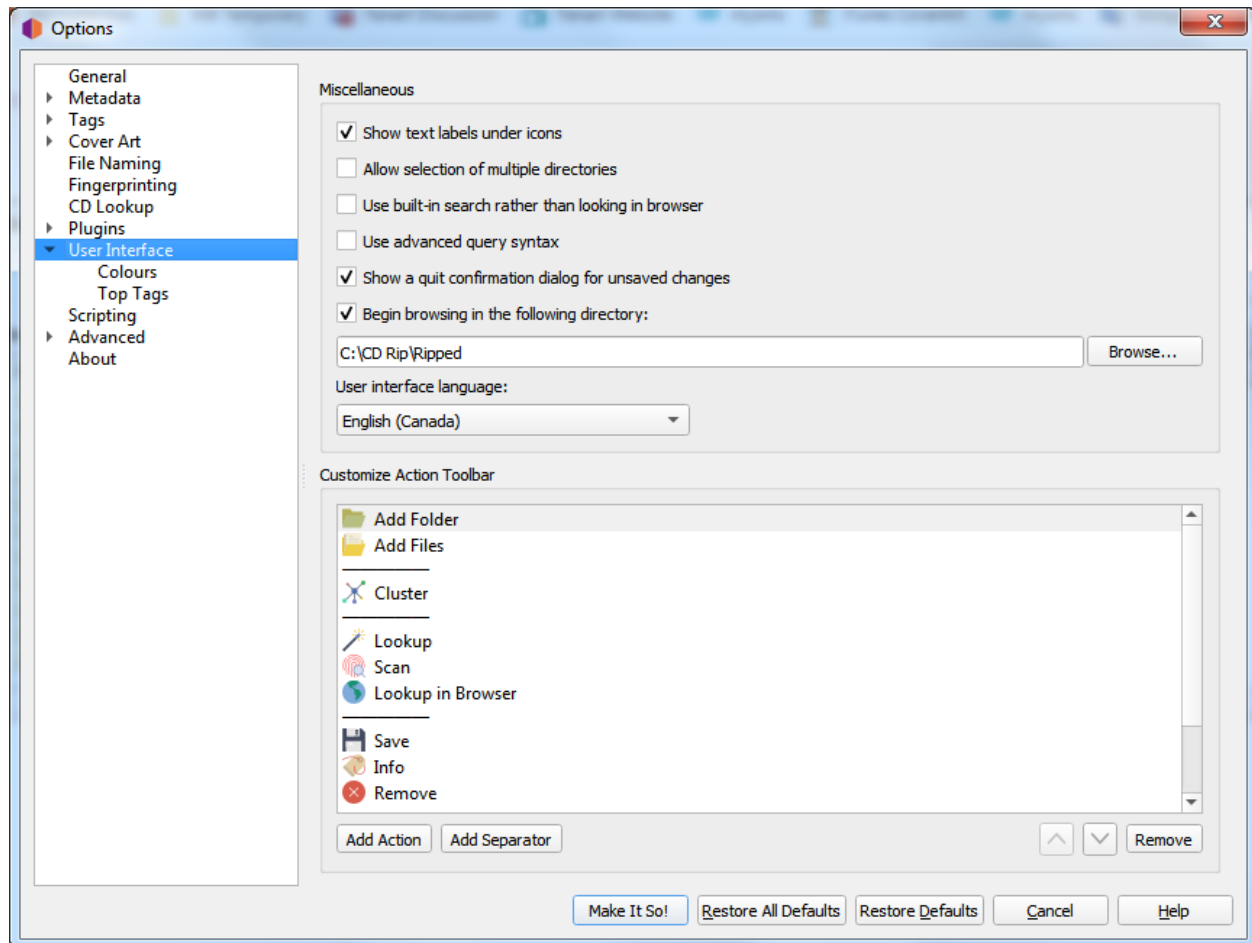
Vous pouvez installer un plugin tiers qui n'apparaît pas dans la liste des plugins, en suivant les étapes suivantes :

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.

### 4.3.9 Options d'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 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.

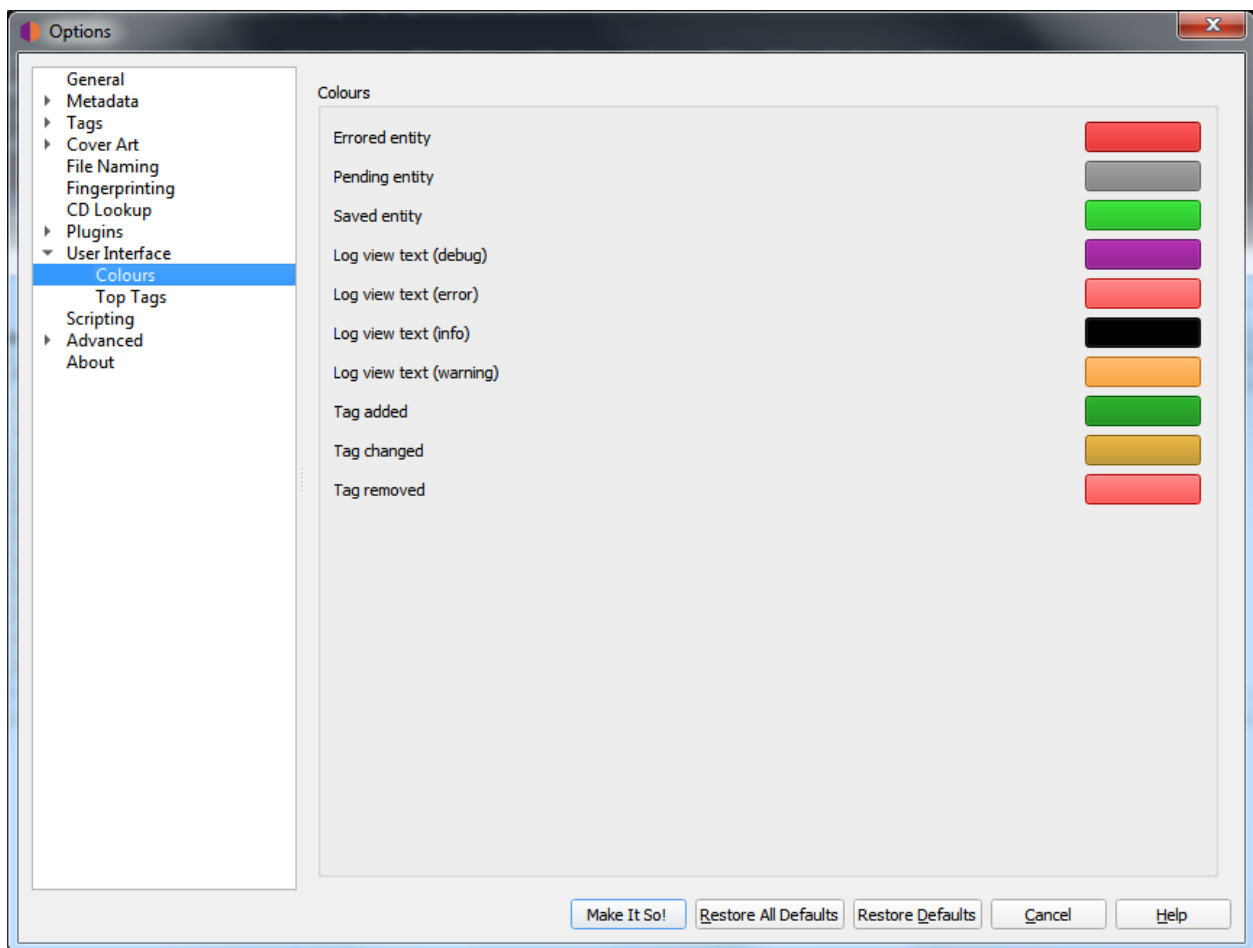
### Langue de l'interface utilisateur

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.

### 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

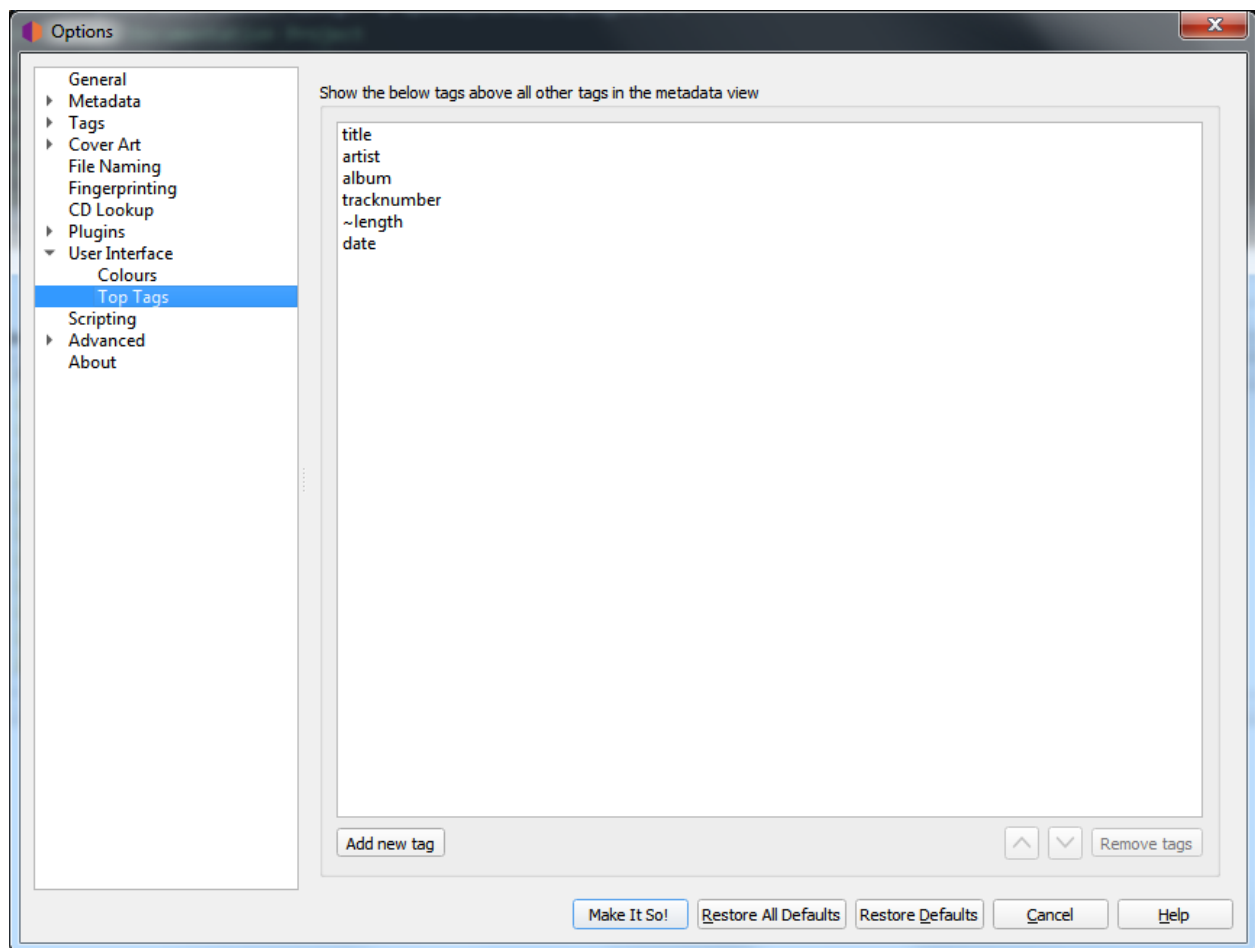


Cette section vous permet de personnaliser les différentes couleurs utilisées dans l'interface utilisateur de Picard. Pour changer une couleur, cliquez simplement sur le bloc de couleur actuellement affiché pour la condition de texte souhaitée pour faire apparaître une boîte de dialogue de sélection,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

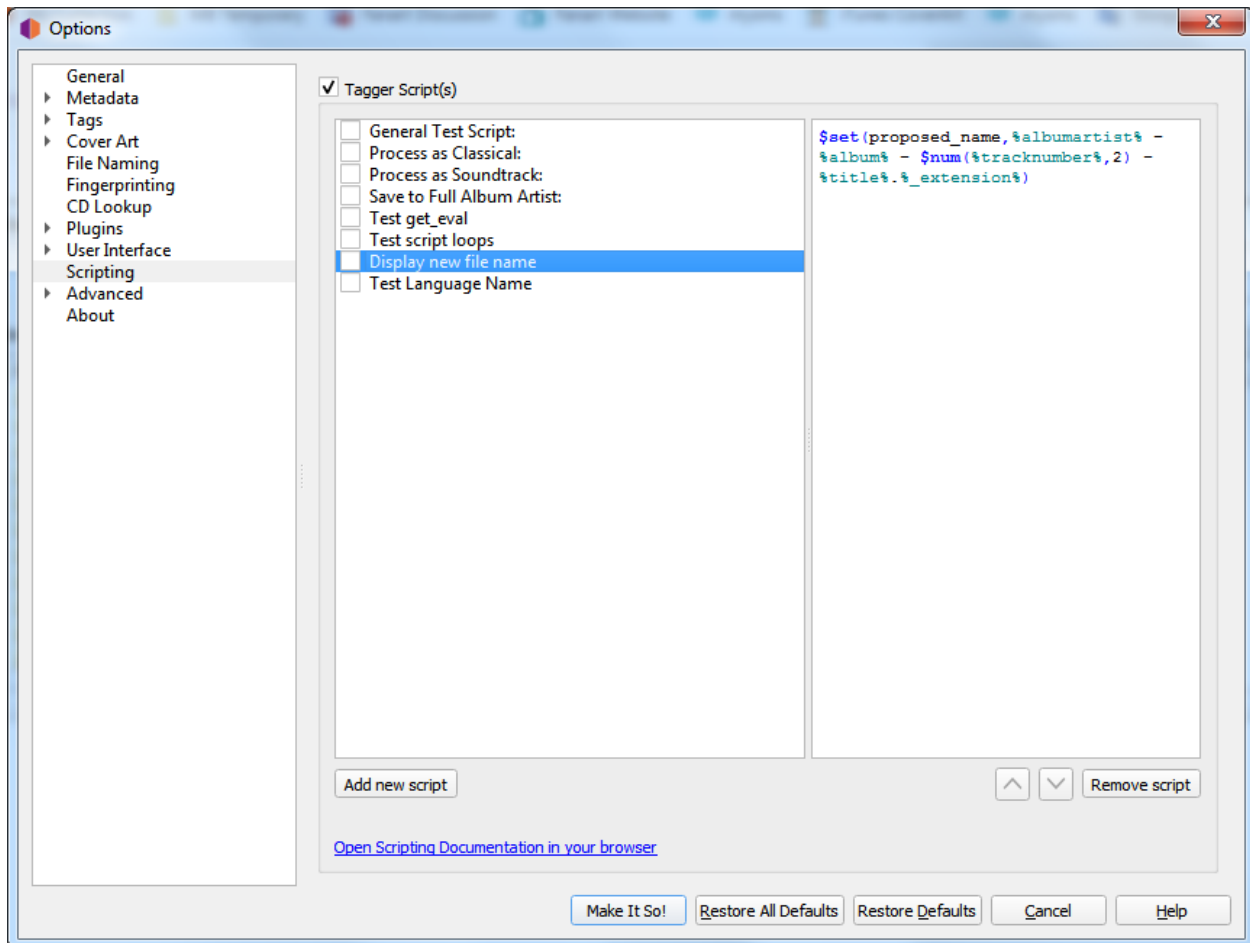
### Top Tags



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.



### 4.3.10 Options de script



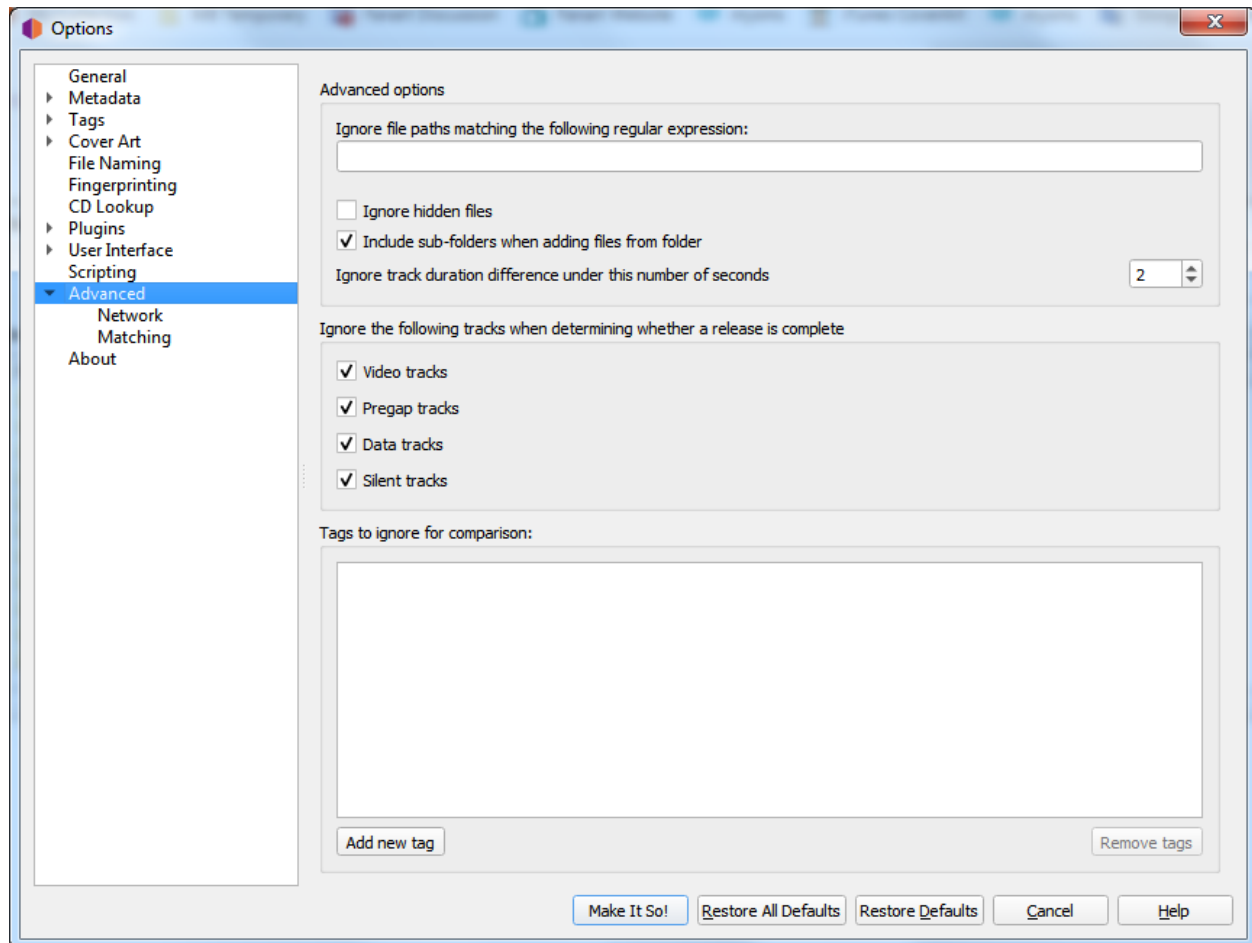
Cette section permet la gestion des scripts de balisage définis par l'utilisateur.

The « Tagger Script(s) » checkbox at the top of the page allows you to completely disable all tagging scripts. This can be useful when tracking down a problem with Picard's configuration.

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.

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

### 4.3.11 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.

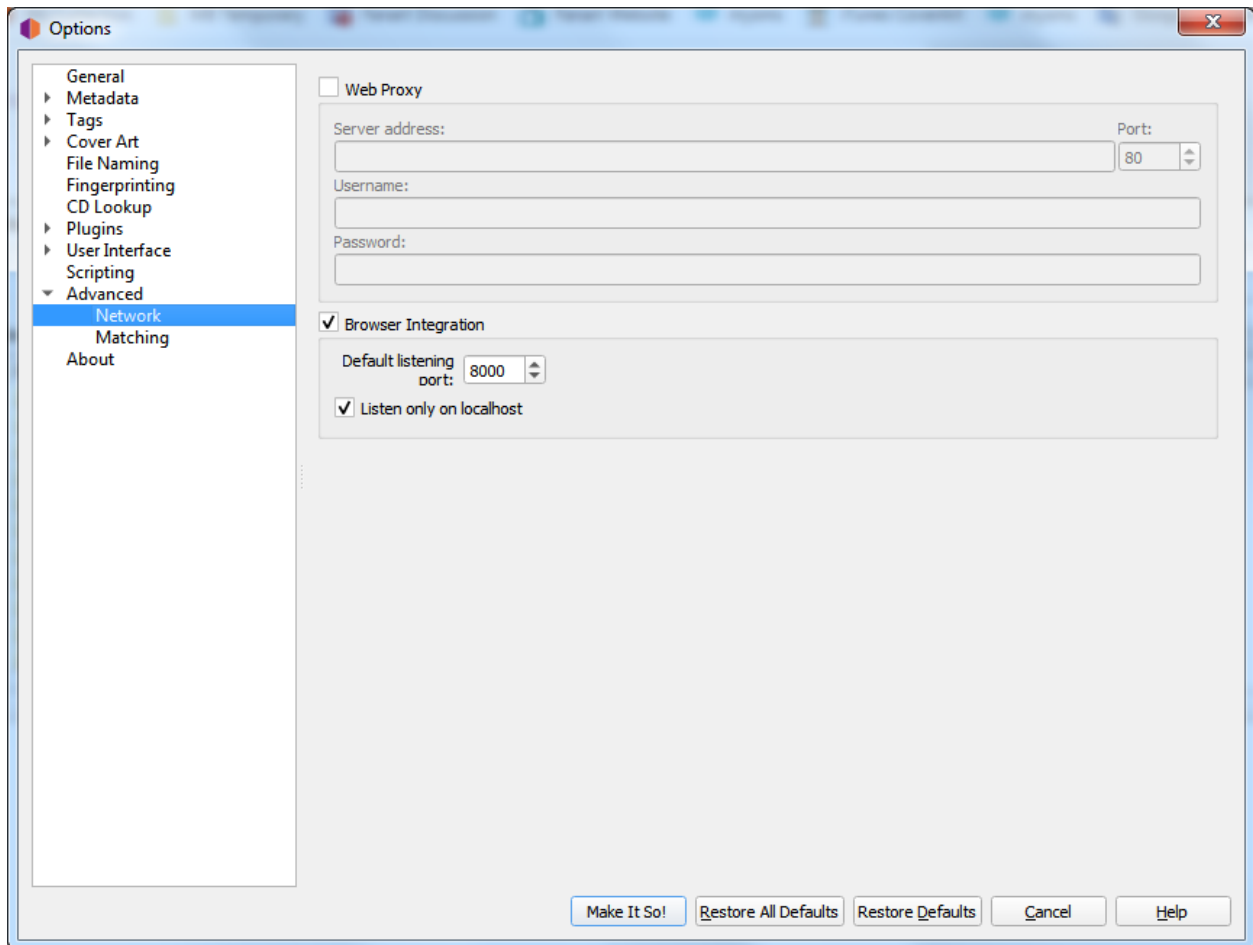
#### Ignorez les pistes suivantes pour déterminer si une version est terminée

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. Les paramètres requis sont **Adresse du serveur**, **Port**, **Nom d'utilisateur** et **Mot de passe**.

### 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](http://musicbrainz.org) dans votre navigateur depuis Picard, le site Web affichera le bouton  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

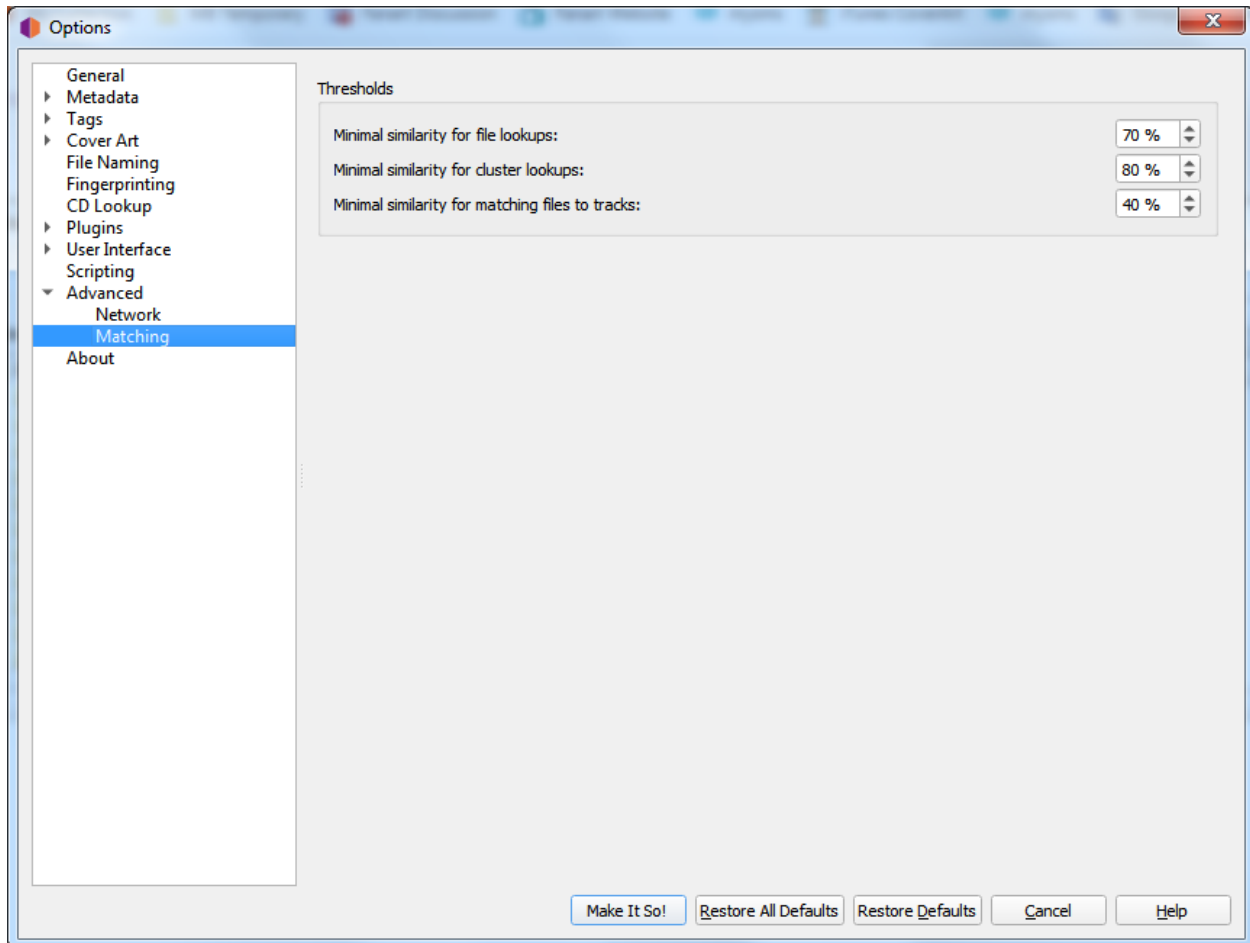
Cela identifie le port par défaut que Picard écouterait pour l'intégration du navigateur. Si le port n'est pas disponible, Picard essaiera d'augmenter le numéro de port de un jusqu'à ce qu'il trouve un port libre.

### 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.

## Matching



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.

## CHAPITRE 5

---

### Balises et 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.

## 5.1 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 page Web [Cartographie des tags Picard](#) pour plus de détails.

---

---

**Note :** Les balises qui reposent sur les informations des fichiers (par exemple : bpm) ne sont disponibles que pour une utilisation dans le script de nommage de fichier.

---

### **acoustid\_fingerprint**

AcoustID Fingerprint pour la piste.

### **acoustid\_id**

AcoustID associé à la piste.

### **album**

Title of the release.

### **albumartist**

Artiste ou artistes principalement crédités sur la sortie.

### **albumartistsort**

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

### **albumsort**

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.

**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 (AAAA-MM-JJ) de la version la plus ancienne du groupe de mise à jour destinée à fournir, par exemple, la date de sortie de la version vinyle de ce que vous avez sur le 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 :** 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 originale prévue pour l'année de sortie de l'enregistrement original.

**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 les variables de base [\\_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.

## 5.2 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 page Web [Cartographie des tags Picard](#) pour plus de détails.

---

### 5.2.1 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)
```

## 5.2.2 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*)

## 5.3 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.

---

---

**Note :** Les variables qui reposent sur les informations des fichiers (par exemple : `_bitrate`) ne sont disponibles que pour une utilisation dans le script de dénomination de fichier.

---

**`_absolutetracknumber`**

Le numéro absolu de cette piste sans tenir compte du numéro de disque (c'est-à-dire : `%_absolutetracknumber%` de `%_totalalbumtracks%`). 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*)

### **\_bitrate**

Le débit binaire approximatif en kbps. Uniquement disponible pour le script de dénomination de fichier.

### **\_bits\_per\_sample**

Le nombre de bits de données par échantillon. Uniquement disponible pour le script de dénomination de fichier.

### **\_channels**

Le nombre de canaux audio dans le fichier. Uniquement disponible pour le script de dénomination de fichier.

### **\_dirname**

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

### **\_extension**

L'extension du fichier. Uniquement disponible pour le script de dénomination de fichier. (*depuis Picard 0.9*)

### **\_filename**

Le nom du fichier sans extension. Uniquement disponible pour le script de dénomination de fichier. (*depuis Picard 1.1*)

### **\_format**

Le format multimédia du fichier (par exemple : MPEG-1 Audio). Uniquement disponible pour le script de dénomination de fichier.

### **\_length**

La longueur de la piste au format mins :secs.

### **\_multiartist**

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

### **\_primaryreleasetype**

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

### **\_rating**

Note 0-5 par les utilisateurs de Musicbrainz.

### **\_releasecomment**

Le commentaire d'homonymie pour la version. (*depuis Picard 0,15*)

### **\_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.

### **\_releaselanguage**

La langue de la version selon [ISO 639-3](#). (*depuis Picard 0,10*)

**\_sample\_rate**

Le nombre d'échantillons de numérisation par seconde (Hz). Uniquement disponible pour le script de dénomination de fichier.

**\_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.

## 5.4 Variables avancées

If you enable tagging with *Advanced Relationships*, you get these extra variables :

---

**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.

## 5.5 Tags 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.

---

## 5.6 Balises des plugins

Plugins de Picard *Les plugins* peuvent ajouter plus de balises. Voici quelques exemples.

### 5.6.1 Plugin Last.fm

**genre**

Pseudo-genre basé sur des balises de folksonomie.

### 5.6.2 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.

---

## 5.7 Les autres informations

Pour des détails techniques sur l'écriture des balises dans les fichiers, voir [Cartographie des tags Picard](#).

Si vous définissez de nouvelles variables, celles-ci seront enregistrées en tant que nouvelles balises dans les fichiers basés sur ID3, APEv2 et VORBIS. Pour les fichiers basés sur ID3, ceux-ci seront enregistrés et rechargés à partir de cadres d'informations textuelles définies par l'utilisateur ID3 (TXXX). Ils ne seront pas enregistrés dans des fichiers ASF ou MP4.

Pour les balises 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 :` ». Actuellement, ces balises ne sont pas chargées dans les variables lorsque vous rechargez le fichier dans Picard. (*depuis Picard 0.9*)

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.

## 6.1 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(...)`).

Pour utiliser des parenthèses ou des virgules telles quelles dans un appel de fonction, vous devez les échapper avec une barre oblique inverse.

## 6.2 Variables de métadonnées

Voir *Balises et 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.

---

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

### 7.1 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 :

#### 7.1.1 \$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 aplatir.

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)
```

(suite sur la page suivante)

(suite de la page précédente)

<code>\$copy (target, source)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$set (source, one)</code>		
<code>\$copy (target, source)</code>	<code>==&gt;</code>	<code>"one"</code>
<code>\$setmulti (source, one)</code>		
<code>\$copy (target, source)</code>	<code>==&gt;</code>	<code>"one"</code>
<code>\$setmulti (source, one; two)</code>		
<code>\$copy (target, source)</code>	<code>==&gt;</code>	<code>"one; two"</code>

### 7.1.2 \$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 aplatir. 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 :

<code>\$set (target, )</code>		
<code>\$set (source, one)</code>		
<code>\$copymerge (target, source)</code>	<code>==&gt;</code>	<code>"one"</code>
<code>\$set (target, zero)</code>		
<code>\$set (source, one)</code>		
<code>\$copymerge (target, source)</code>	<code>==&gt;</code>	<code>"zero; one"</code>
<code>\$set (target, zero)</code>		
<code>\$setmulti (source, one; two)</code>		
<code>\$copymerge (target, source)</code>	<code>==&gt;</code>	<code>"zero; one; two"</code>
<code>\$setmulti (target, zero; two)</code>		
<code>\$setmulti (source, one; two)</code>		
<code>\$copymerge (target, source)</code>	<code>==&gt;</code>	<code>"zero; two; one"</code>
<code>\$set (target, zero; one; zero)</code>		
<code>\$set (source, one; two; three)</code>		
<code>\$copymerge (target, source)</code>	<code>==&gt;</code>	<code>"zero, one; two; three"</code>

### 7.1.3 \$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
```

### 7.1.4 \$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 :** To create a variable which can be used for the file naming string, but which will not be written as a tag in the file, prefix the variable name with an underscore. `%something%` will create a « something » tag; `_%something%` will not.

---

#### 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
```



### 7.1.5 \$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 :

<code>\$setmulti(genre,\$lower(%genre%))</code>	<code>==&gt;</code>	all "genre" elements in lower case
<code>\$setmulti(alpha,A; B; C)</code>	<code>==&gt;</code>	3 elements ("A", "B" and "C")
<code>\$setmulti(mixed,A:A; B:B, :)</code>	<code>==&gt;</code>	3 elements ("A", "A; B" and "B")

### 7.1.6 \$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 :

<code>\$unset(performer:*)</code>
-----------------------------------

## 7.2 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 :

### 7.2.1 \$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 :

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

### 7.2.2 \$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)	==>	" "

(suite sur la page suivante)

(suite de la page précédente)

<code>\$find(, a)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$find(abcdef, )</code>	<code>==&gt;</code>	<code>"1"</code>

### 7.2.3 \$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 :

<code>\$firstalphachar(abc)</code>	<code>==&gt;</code>	<code>"A"</code>
<code>\$firstalphachar(123)</code>	<code>==&gt;</code>	<code>"#"</code>
<code>\$firstalphachar(***)</code>	<code>==&gt;</code>	<code>"#"</code>
<code>\$firstalphachar(***, )</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$firstalphachar(***, ^)</code>	<code>==&gt;</code>	<code>"^"</code>
<code>\$firstalphachar(***, non-alpha)</code>	<code>==&gt;</code>	<code>"non-alpha"</code>

### 7.2.4 \$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)   ==> ""
```

### 7.2.5 \$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)
$set(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)
```

### 7.2.6 \$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"
```

### 7.2.7 \$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) ==>  ""
```

### 7.2.8 \$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)
$len(%foo%)  ==>  "3"

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

### 7.2.9 \$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"
```

### 7.2.10 \$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) ==> "000000000000000000123"
$num(123,50) ==> "000000000000000000123"
$num(123,5.5) ==> ""
$num(1.23,10) ==> "0000000000"
```

### 7.2.11 \$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 remplie 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, +)    ==>  "++abc"
$pad(abc, 0, +)    ==>  "abc"
$pad(abc, 5, )     ==>  "abc"
$pad(abc, 5, XY)   ==>  "XYXYabc" (note final length is incorrect)
$pad(abc, , +)     ==>  ""
$pad(abc, x, +)    ==>  ""
```

### 7.2.12 \$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)
$replace(%foo%,%bar%,dog)         ==>  "I like dogs the best"

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

(suite sur la page suivante)

(suite de la page précédente)

```
$replace(I like cats the best,cat,dog) ==> "I like dogs the best"
$replace(I like cats the best,pig,dog) ==> "I like cats the best"
$replace(I like cats the best,cat,) ==> "I like s the best"
$replace(Bad replace,,_) ==> "_B_a_d_ _r_e_p_l_a_c_e_"
```

### 7.2.13 \$reverse

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

Catégorie : texte

#### La description :

Renvoie `texte` dans l'ordre inverse.

#### Exemple :

Les instructions suivantes renverront les valeurs indiquées :

```
$set(foo,abcde)
$reverse(%foo%) ==> "edcba"

$reverse(abcde) ==> "edcba"
```

### 7.2.14 \$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) ==> ""
```



### 7.2.15 \$rreplace

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

Catégorie : texte

#### La description :

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

#### Exemple :

Les instructions suivantes renverront les valeurs indiquées :

<code>\$rreplace(test \ (disc 1\), \\s\\ (disc \\d+\\), )</code>	<code>==&gt;</code>	<code>"test "</code>
<code>\$rreplace(test, [t, )</code>	<code>==&gt;</code>	<code>"test "</code>

### 7.2.16 \$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](#).

#### Exemple :

Les instructions suivantes renverront les valeurs indiquées :

<code>\$rsearch(test \ (disc 1\), \\ (disc \\ (\\d+\\) \\))</code>	<code>==&gt;</code>	<code>"1"</code>
<code>\$rsearch(test \ (disc 1\), \\ (disc \\d+\\))</code>	<code>==&gt;</code>	<code>" (disc 1) "</code>
<code>\$rsearch(test, x)</code>	<code>==&gt;</code>	<code>" "</code>

### 7.2.17 \$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.)
$strip( This text has been stripped. )
$strip(This text has been stripped.)
$strip(This text\rhas\nbeen\tstripped.)
```

## 7.2.18 \$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)  ==> ""
```

### 7.2.19 \$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'album `albumartiste` est « Le Butcherettes », les instructions suivantes renverront les valeurs indiquées :

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

### 7.2.20 \$title

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

Catégorie : texte

Mis en œuvre : Picard 2.1

#### La description :

Renvoie `texte` dans la casse du titre (premier caractère de chaque mot en majuscule).

#### Exemple :

L'instruction suivante renverra la valeur indiquée :

<code>\$title(tHe houR is upOn uS)</code>	<code>==&gt;</code>	<code>"The Hour Is Upon Us"</code>
---	---------------------	------------------------------------

### 7.2.21 \$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"
```

## 7.2.22 \$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)   ==> ""
```

## 7.2.23 \$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"
```

## 7.3 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 :

### 7.3.1 \$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 la 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)  ==> ""
```

### 7.3.2 \$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 « ; » s'il n'est pas passé) pour contraindre la valeur à une valeur multiple appropriée variable.

#### Exemple :

Les instructions suivantes renverront les valeurs indiquées :

<code>\$set(foo,First:A; Second:B)</code>	
<code>\$join(%foo%, &gt;&gt; )</code>	<code>==&gt; "First:A; Second:B"</code>
<code>\$join(%foo%, &gt;&gt; ,:)</code>	<code>==&gt; "First &gt;&gt; A; Second &gt;&gt; B"</code>
<code>\$setmulti(bar,First:A; Second:B)</code>	
<code>\$join(%bar%, &gt;&gt; )</code>	<code>==&gt; "First:A &gt;&gt; Second:B"</code>
<code>\$join(%bar%, &gt;&gt; ,:)</code>	<code>==&gt; "First &gt;&gt; A; Second &gt;&gt; B"</code>
<code>\$join(First:A; Second:B,)</code>	<code>==&gt; "First:ASecond:B"</code>
<code>\$join(First:A; Second:B, &gt;&gt; )</code>	<code>==&gt; "First:A &gt;&gt; Second:B"</code>
<code>\$join(First:A; Second:B, &gt;&gt; ,:)</code>	<code>==&gt; "First &gt;&gt; A; Second &gt;&gt; B"</code>

### 7.3.3 \$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>	<code>==&gt; "0"</code>

(suite sur la page suivante)

(suite de la page précédente)

```

$set(foo,A; B; C)
$lenmulti(%foo%)           ==>  "1"

$setmulti(foo,A; B; C)
$lenmulti(%foo%)           ==>  "3"

$lenmulti(A; B; C)          ==>  "3"
$lenmulti(A:A; B:B; C:C, :) ==>  "4"
$lenmulti(,)                ==>  "0"
$lenmulti(, :)              ==>  "0"
$lenmulti(A; B; C, )        ==>  "1"

```

### 7.3.4 \$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.

**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%,

```

(suite sur la page suivante)

(suite de la page précédente)

<code>\$upper(%_loop_count%=%_loop_value%)</code>	<code>==&gt;</code>	<code>"1=FIRST:A; 2=SECOND:B"</code>
<code>\$map(%bar%,</code> <code>    <code>\$upper(%_loop_count%=%_loop_value%), :</code></code>	<code>==&gt;</code>	<code>"1=FIRST:2=A; SECOND:3=B"</code>
<code>\$map(First:A; Second:B,</code> <code>    <code>\$upper(%_loop_count%=%_loop_value%)</code></code>	<code>==&gt;</code>	<code>"1=FIRST:A; 2=SECOND:B"</code>

### 7.3.5 \$performer

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

Catégorie : multi-valeur

Mis en œuvre : Picard 0.10

#### La description :

Returns the performers where the performance type matches `pattern` separated by `separator` (or a comma followed by a space « , » if not passed). If `pattern` is blank, then all performers will be returned. Note that the `pattern` to be matched is case-sensitive.

#### Exemple :

Avec les balises performer comme `performer:guitar = « Ann »`, `performer:rhythm-guitar = « Bob »` et `performer:drums = « Cindy »`, les instructions suivantes renverront les valeurs indiqué :

<code>\$set(foo, guitar)</code>	
<code>\$performer(%foo%)</code>	<code>==&gt; "Ann, Bob"</code>
<code>\$performer(guitar)</code>	<code>==&gt; "Ann, Bob"</code>
<code>\$performer(Guitar)</code>	<code>==&gt; ""</code>
<code>\$performer(rhythm-guitar)</code>	<code>==&gt; "Bob"</code>
<code>\$performer()</code>	<code>==&gt; "Ann, Bob, Cindy"</code>
<code>\$performer(, / )</code>	<code>==&gt; "Ann / Bob / Cindy"</code>

### 7.3.6 \$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"
```

### 7.3.7 \$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)
$slice(%foo%,1)          ==>  "B; C; D; E"
```

(suite sur la page suivante)

(suite de la page précédente)

<code>\$slice(A; B; C; D; E, 1)</code>	<code>==&gt;</code>	<code>"B; C; D; E"</code>
<code>\$slice(A; B; C; D; E, 1, 3)</code>	<code>==&gt;</code>	<code>"B; C"</code>
<code>\$slice(A; B; C; D; E, , 3)</code>	<code>==&gt;</code>	<code>"A; B; C"</code>
<code>\$slice(A; B; C; D; E, 1, 3)</code>	<code>==&gt;</code>	<code>"B; C"</code>
<code>\$slice(A; B; C; D; E, 1, -1)</code>	<code>==&gt;</code>	<code>"B; C; D"</code>
<code>\$slice(A; B; C; D; E, -4, 4)</code>	<code>==&gt;</code>	<code>"B; C; D"</code>
<code>\$slice(A:A; B:B; C:C; D:D; E:E, , 1, :)</code>	<code>==&gt;</code>	<code>"A"</code>
<code>\$slice(A:A; B:B; C:C; D:D; E:E, -2, , :)</code>	<code>==&gt;</code>	<code>"D; E:E"</code>
<code>\$slice(A:A; B:B; C:C; D:D; E:E, 2, 4, :)</code>	<code>==&gt;</code>	<code>"B; C:C; D"</code>

### 7.3.8 \$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 «`;` » s'il n'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 :

<code>\$set(foo,B; C; E; D; A)</code>	
<code>\$sortmulti(%foo%)</code>	<code>==&gt; "B; C; E; D; A"</code>
<code>\$setmulti(foo,B; C; E; D; A)</code>	
<code>\$sortmulti(%foo%)</code>	<code>==&gt; "A; B; C; D; E"</code>
<code>\$sortmulti(B; D; E; A; C)</code>	<code>==&gt; "A; B; C; D; E"</code>
<code>\$sortmulti(B:AB; D:C; E:D; A:A; C:X, :)</code>	<code>==&gt; "A; C:AB; D:B:C; E:D; A:X"</code>
<code>\$sortmulti(,)</code>	<code>==&gt; ""</code>
<code>\$sortmulti(, :)</code>	<code>==&gt; ""</code>

## 7.4 Fonctions mathématiques

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

### 7.4.1 \$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 :

<code>\$add(20, 15)</code>	<code>==&gt;</code>	<code>"35"</code>
<code>\$add(20, -15)</code>	<code>==&gt;</code>	<code>"5"</code>
<code>\$add(20, 14, 1)</code>	<code>==&gt;</code>	<code>"35"</code>
<code>\$add(20, 10, 3, 2)</code>	<code>==&gt;</code>	<code>"35"</code>
<code>\$add(20, 10, 3, )</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$add(20, 10, 3, a)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$add(20, 10, 3.5)</code>	<code>==&gt;</code>	<code>" "</code>

### 7.4.2 \$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 :

<code>\$div(10, 3)</code>	<code>==&gt;</code>	<code>"3"</code>
<code>\$div(10, -3)</code>	<code>==&gt;</code>	<code>"-4"</code>
<code>\$div(-10, 3)</code>	<code>==&gt;</code>	<code>"-4"</code>
<code>\$div(10, 3, 2)</code>	<code>==&gt;</code>	<code>"1"</code>
<code>\$div(10, -3, -2)</code>	<code>==&gt;</code>	<code>"2"</code>
<code>\$div(10, 2, 1.5)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$div(10, 2, 0)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$div(10, 2, x)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$div(10, 2, )</code>	<code>==&gt;</code>	<code>" "</code>

### 7.4.3 \$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 :

<code>\$mod(0, 3)</code>	<code>==&gt;</code>	<code>"0"</code>
<code>\$mod(10, 3)</code>	<code>==&gt;</code>	<code>"1"</code>
<code>\$mod(10, -3)</code>	<code>==&gt;</code>	<code>"-2"</code>
<code>\$mod(-13, 10)</code>	<code>==&gt;</code>	<code>"7"</code>
<code>\$mod(13, -10)</code>	<code>==&gt;</code>	<code>"-7"</code>
<code>\$mod(10, 3, 1)</code>	<code>==&gt;</code>	<code>"0"</code>
<code>\$mod(50, 17, 9)</code>	<code>==&gt;</code>	<code>"7"</code>
<code>\$mod(51, 3, 0)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$mod(51, a)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$mod(a, 10)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$mod(, 10)</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$mod(10, )</code>	<code>==&gt;</code>	<code>" "</code>
<code>\$mod(10, 3.5)</code>	<code>==&gt;</code>	<code>" "</code>

### 7.4.4 \$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 :

<code>\$mul(1, 2)</code>	<code>==&gt;</code>	<code>"2"</code>
<code>\$mul(1, 2, 3)</code>	<code>==&gt;</code>	<code>"6"</code>
<code>\$mul(1, 2, 0)</code>	<code>==&gt;</code>	<code>"0"</code>
<code>\$mul(1, -2, 3)</code>	<code>==&gt;</code>	<code>"-6"</code>
<code>\$mul(-1, 2, -3)</code>	<code>==&gt;</code>	<code>"6"</code>
<code>\$mul(1, 2, )</code>	<code>==&gt;</code>	<code>" "</code>

(suite sur la page suivante)

(suite de la page précédente)

```
$mul (1, 2, x)      ==> " "
$mul (1, 2.5)      ==> " "
```

### 7.4.5 \$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) ==> " "
```

## 7.5 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 :

### 7.5.1 \$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)
```

## 7.5.2 \$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)
```

## 7.5.3 \$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)

```

### 7.5.4 \$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)

```

### 7.5.5 \$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)

```

### 7.5.6 \$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 :

<code>\$gt (-1, 0)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gt (0, 0)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gt (1, 0)</code>	<code>==&gt;</code>	<code>"1"</code>	<code>(True)</code>
<code>\$gt (, )</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gt (, 0)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gt (0, )</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gt (a, 1)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gt (1, a)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gt (1, 1.5)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>

### 7.5.7 \$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 :

<code>\$gte (-1, 0)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gte (0, 0)</code>	<code>==&gt;</code>	<code>"1"</code>	<code>(True)</code>
<code>\$gte (1, 0)</code>	<code>==&gt;</code>	<code>"1"</code>	<code>(True)</code>
<code>\$gte (, )</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gte (, 0)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gte (0, )</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gte (a, 1)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gte (1, a)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>
<code>\$gte (1, 1.5)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(False)</code>



### 7.5.8 \$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)            ==> ""
$if(,True,False)                  ==> "False"
$if(,True,False)                  ==> "True"
$if(,$set(value,True),$set(value,False)) ==> Sets "value" to "False"
$set(value,$if(%bar%,True,False)) ==> Sets "value" to "False"
```

### 7.5.9 \$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 :

```
$set(foo,)
$set(bar,Something)
$if2(%foo%,%bar%,Three)           ==> "Something"
$if2(,%bar%,Three)                ==> "Something"
$if2(,%foo%,%bar%,Three)          ==> "Something"
$if2(%foo%,,%bar%,Three)          ==> ""
$if2(%foo%.,%bar%,Three)          ==> "."
$if2(%foo%,,Three)                ==> "Three"
$if2(%foo%,,,)                    ==> ""
```

### 7.5.10 \$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 :

```
$set (foo, ABCDEFG)
$set (bar, CDE)
$in (%foo%, %bar%) ==> "1" (True)
$in (ABCDE, CDE) ==> "1" (True)
$in (ABCDE, CE) ==> "" (False)
$in (ABCDE, cde) ==> "" (False)
$in (ABCDE, ) ==> "1" (True)
$in (, ) ==> "1" (True)
```

### 7.5.11 \$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)
```

### 7.5.12 \$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)
```

### 7.5.13 \$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)
```

### 7.5.14 \$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)
```

### 7.5.15 \$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(,) ==> ""  (False)
$lte(,0) ==> ""  (False)
$lte(0,) ==> ""  (False)
$lte(a,1) ==> ""  (False)
$lte(1,a) ==> ""  (False)
$lte(1,1.5) ==> "" (False)
```

### 7.5.16 \$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)
```

(suite sur la page suivante)

(suite de la page précédente)

```

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

```

### 7.5.17 \$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)

```

### 7.5.18 \$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 :

Les instructions suivantes renverront les valeurs indiquées :

```

$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)

```

### 7.5.19 \$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 :**

Les instructions suivantes renverront les valeurs indiquées :

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

### 7.5.20 \$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>==&gt;</code>	<code>"1"</code>	<code>(False)</code>
<code>\$not(x)</code>	<code>==&gt;</code>	<code>" "</code>	<code>(True)</code>
<code>\$not( )</code>	<code>==&gt;</code>	<code>" "</code>	<code>(True)</code>
<code>\$not()</code>	<code>==&gt;</code>	<code>Error</code>	

### 7.5.21 \$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)

### 7.5.22 \$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)

## 7.6 Fonctions d'information

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

### 7.6.1 \$datetime

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

Catégorie : information

Mis en œuvre : Picard 2.3

#### La description :

Returns the current date and time in the specified format, which is based on the standard Python `strftime` [format codes](#). If no format is specified the date and time will be returned in the form “2020-02-15 14:26:32”. Note that any special characters such as “%”, “\$”, “(”, “)” and “\” will need to be escaped as shown in the examples below.

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

### 7.6.2 \$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)
```

## 7.7 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 :

### 7.7.1 \$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
      track as 'artist_1', 'artist_2', etc. )
$foreach(%artists%,$set(artist_%_loop_count%,_%_loop_value%))
```

### 7.7.2 \$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...))
```

## 7.8 Fonctions diverses

Les diverses fonctions de script sont :

### 7.8.1 \$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)
```

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

### 8.1 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).

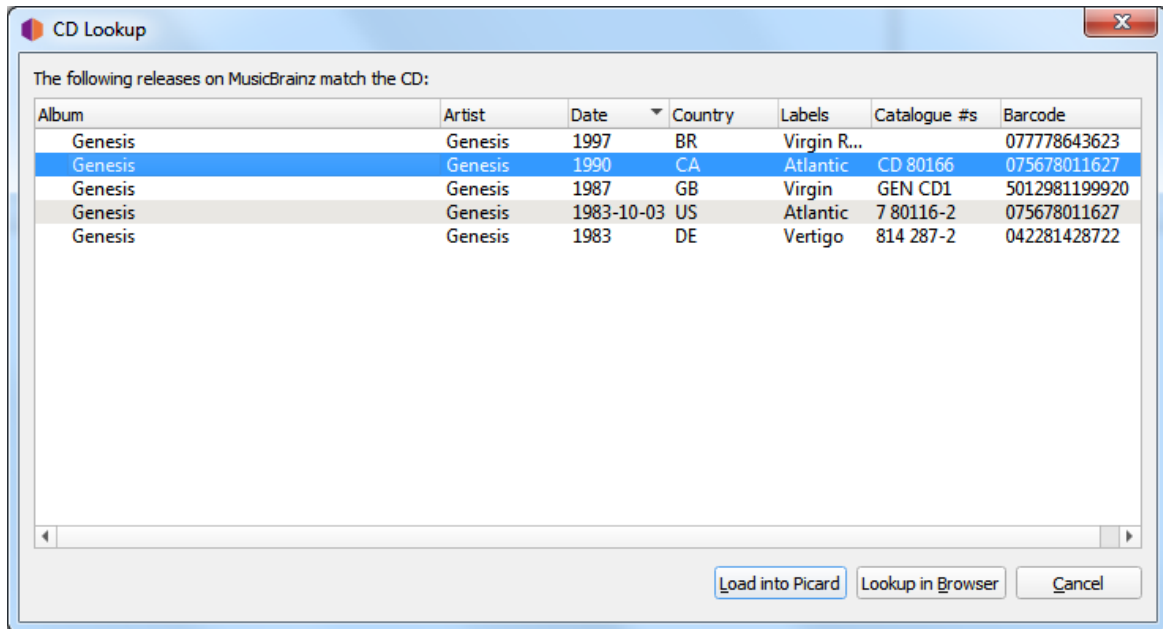
Il existe essentiellement quatre méthodes principales utilisées pour récupérer les informations d'album à partir de la base de données MusicBrainz.

#### 8.1.1 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.

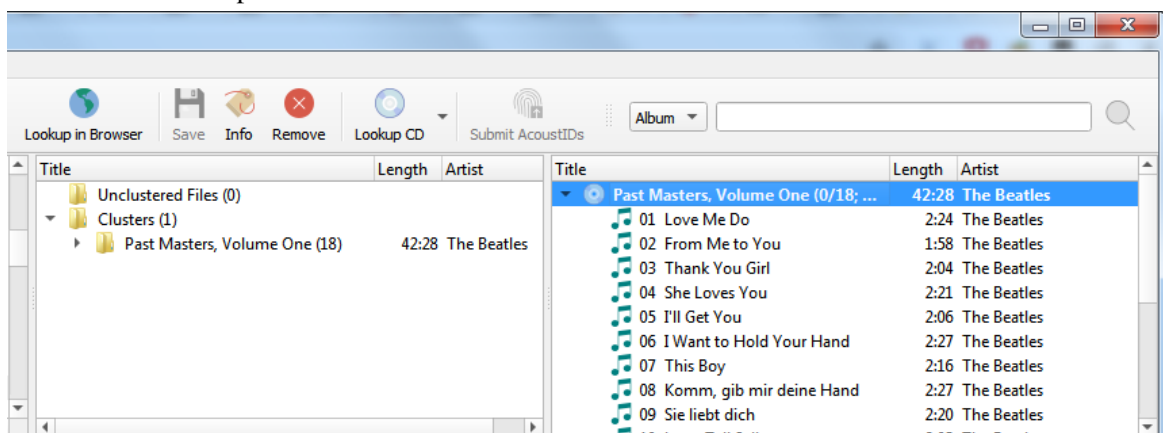
Les étapes à suivre pour rechercher un CD sont :

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.



- 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.



- 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.

**MusicBrainz** rdswift My Data Search Artist Q

About Us Products Search Editing Documentation English R5

### Lookup CD

**Matching CDs**

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	TAGGER
1/1	Genesis	Genesis	CD	BR 1997	Virgin Records America (DO NOT USE! please refer to either "Virgin" or "Virgin America")		077778643623	TAGGER
1/1	Genesis	Genesis	CD	US 1983-10-03	Atlantic (Warner Music imprint)	7 80116-2	075678011627	TAGGER
1/1	Genesis	Genesis	CD	GB 1987	Virgin (worldwide imprint of Virgin Records Ltd. and all its subsidiaries)	GEN CD1	5012981199920	TAGGER
1/1	Genesis	Genesis	CD	DE 1983	Vertigo (British rock label)	814 287-2	042281428722	TAGGER

We used DiscID 31xF1vanXbkVDevm4jXtvX4b91s- to look up this information.

**Search by artist**

Artist:

Search

**Search by release**

Release:

Search

[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 **TAGGER** pour charger les informations pour une publication dans Picard. De plus, vous pouvez sélectionner la version et attacher la toc.

CD 1 (show tracklist)

Genesis	Genesis	JP	1985-06-01	Vertigo (British rock label)	32PD-17	4988011303999	TAGGER
CD 1 (show tracklist)	Genesis	NL	1983-10-03	Mercury Records (or just "Mercury". A UMG imprint, do not use it for ©/® credits)	832 178-1	042281428715	TAGGER
12" Vinyl 1 (show tracklist)	Genesis	CA	1990	Atlantic (Warner Music imprint)	CD 80166	075678011627	TAGGER
CD 1 (show tracklist)	Genesis	XE	-	Atlantic (Warner Music imprint)	7 80116-2, 814287-2	[none]	TAGGER
CD 1 (show tracklist)	Genesis	US	-				TAGGER

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

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**

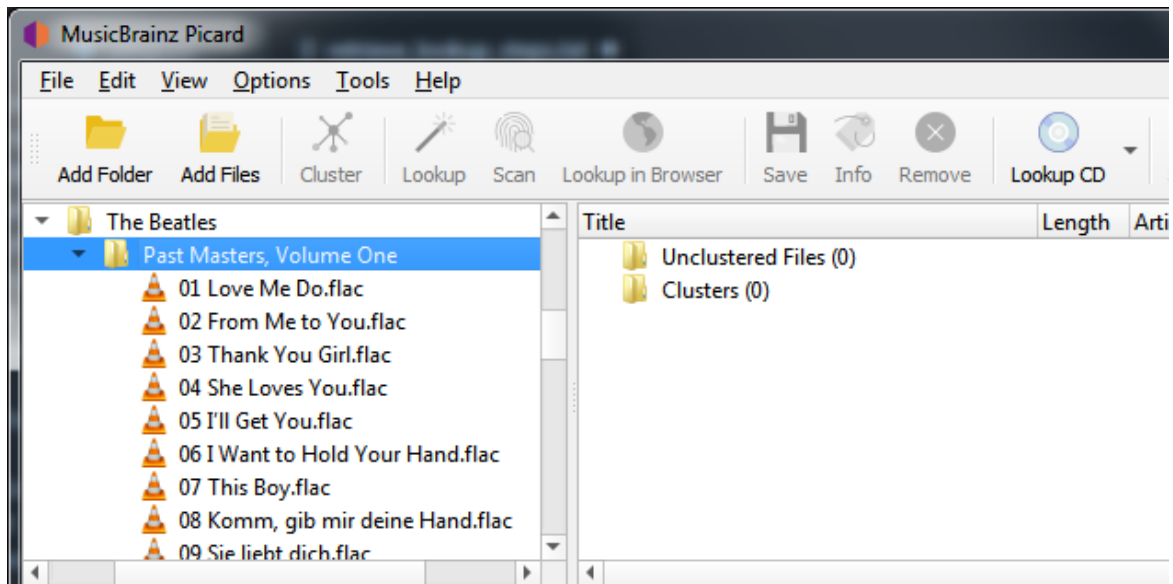
[Donate](#) | [Wiki](#) | [Forums](#) | [IRC](#) | [Bug Tracker](#) | [Blog](#) | [Twitter](#) | [Use beta site](#)

### 8.1.2 Rechercher des fichiers

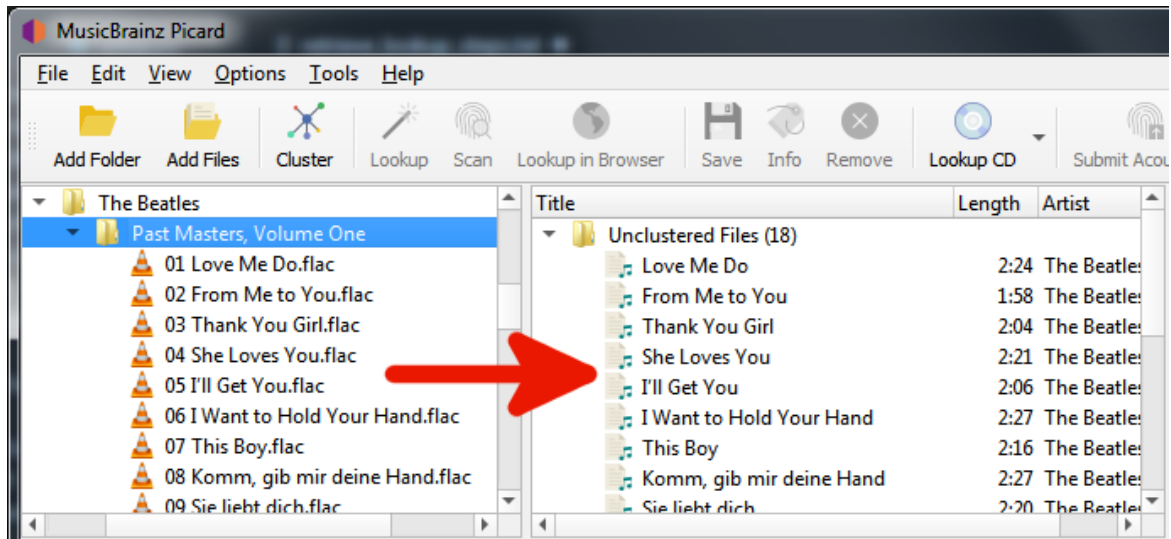
Si vous ne disposez pas du CD et que vos fichiers sont regroupés par album, il s'agit de la méthode préférée pour identifier automatiquement l'album à récupérer. Cela se fait en regroupant les fichiers dans des groupes d'albums dans Picard, puis en effectuant la recherche. Picard essaiera de faire correspondre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

Les étapes à suivre pour rechercher des fichiers sont :

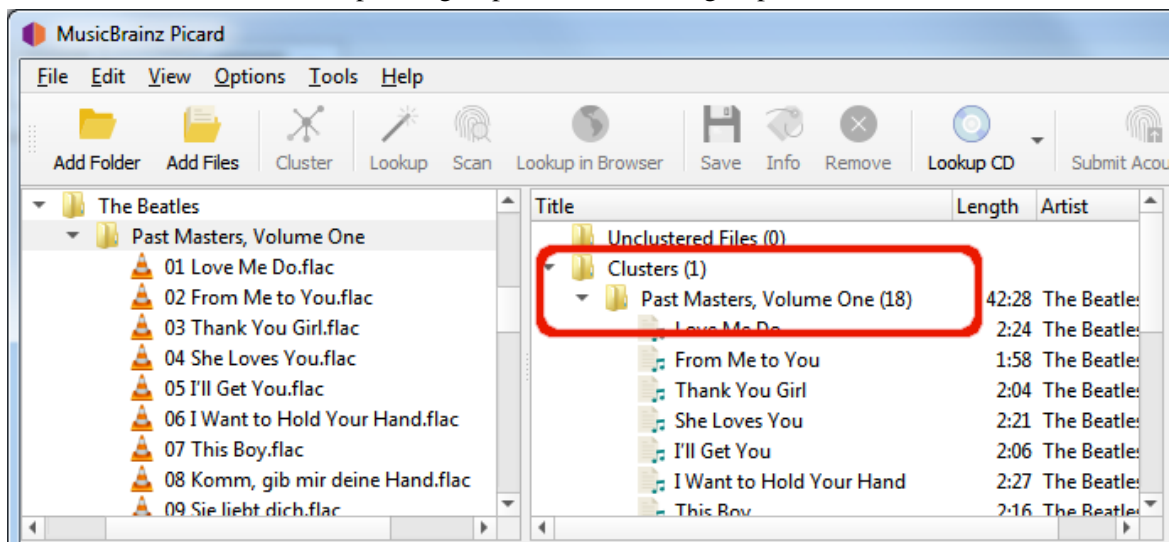
1. Ajoutez vos fichiers en utilisant « *Fichiers → Ajouter des fichiers...* » ou « *Fichiers → Ajouter un dossier...* ». Pour faciliter l'utilisation, il est recommandé d'utiliser le navigateur de fichiers interne pour gérer les interactions du système de fichiers. Ceci est activé à partir de « *Affichage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



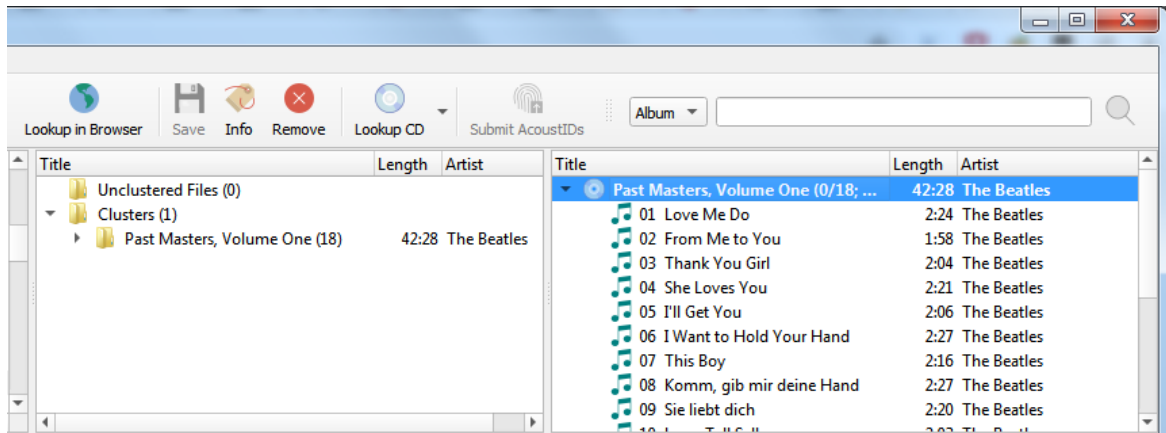
2. Faites glisser le répertoire ou les fichiers sélectionnés vers le dossier « Fichiers non groupés » et attendez que Picard traite les fichiers - les noms passeront du gris au noir.



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



4. 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.

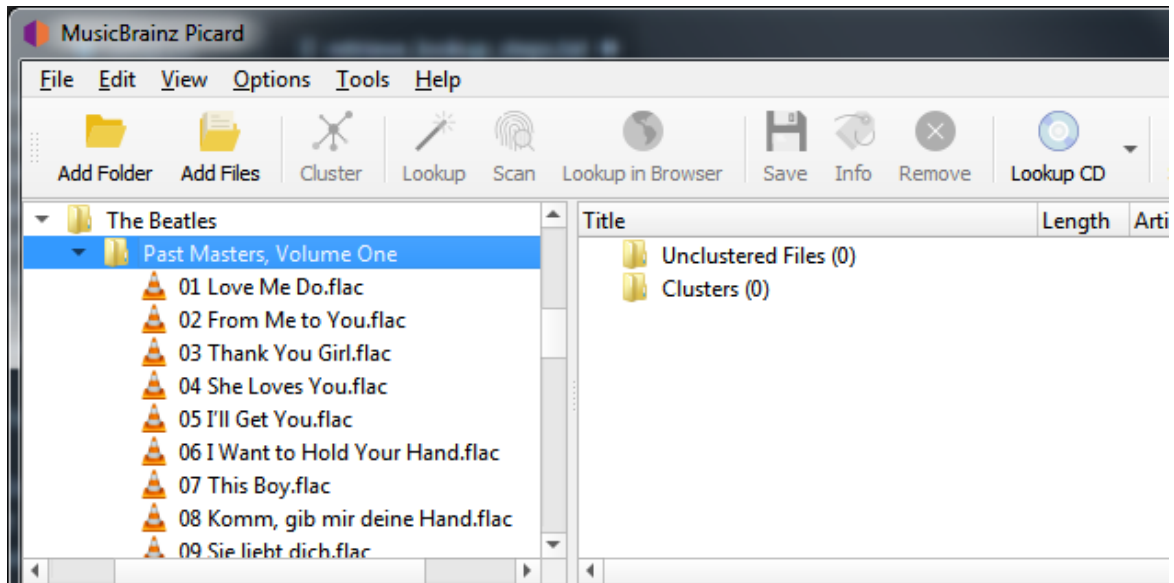
### 8.1.3 Analyser les fichiers

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. 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.

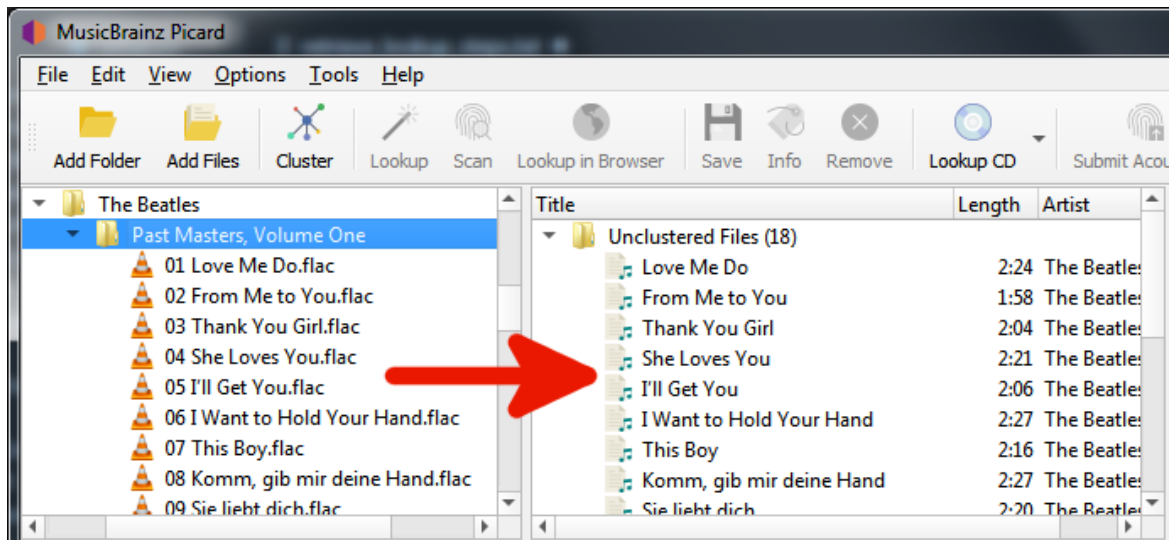
Les étapes à suivre pour analyser et rechercher des fichiers sont les suivantes :

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* ».

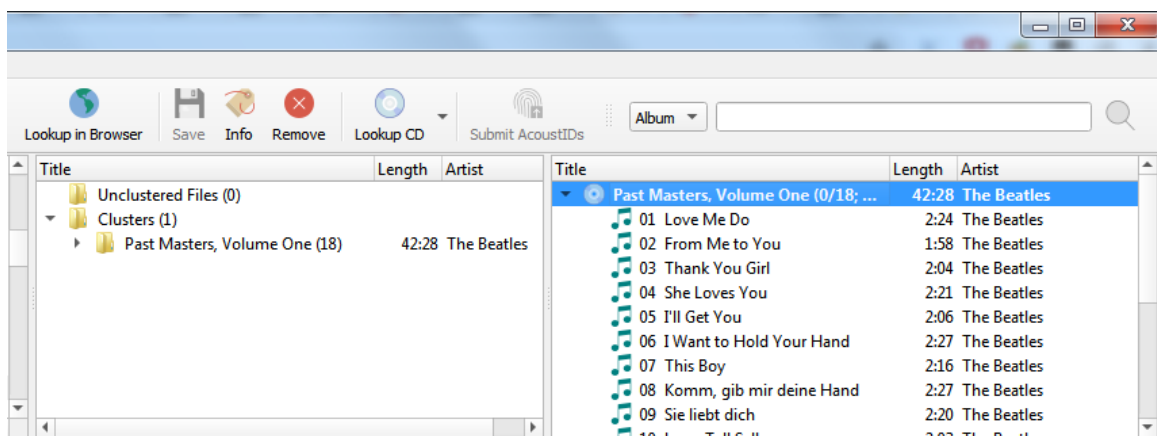




2. Faites glisser le répertoire ou les fichiers sélectionnés vers le dossier « Fichiers non groupés » et attendez que Picard traite les fichiers - les noms passeront du gris au noir.



3. 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.



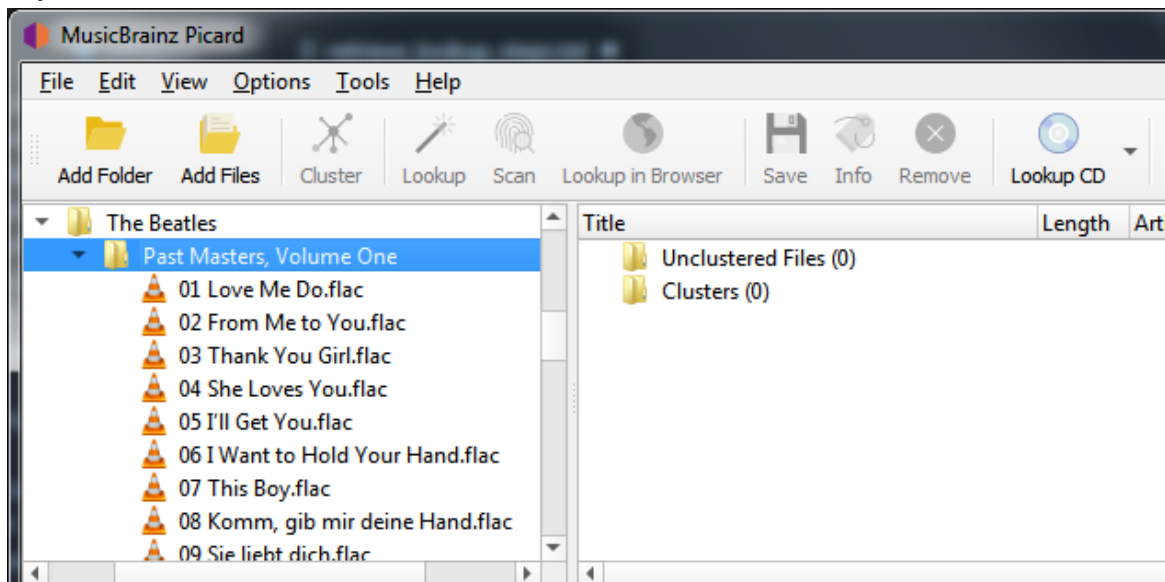
Si aucun album n'a été récupéré ou si l'album récupéré était incorrect, vous devrez peut-être essayer une autre méthode, telle que le regroupement des fichiers ou une recherche dans le navigateur.

### 8.1.4 Recherche dans le navigateur

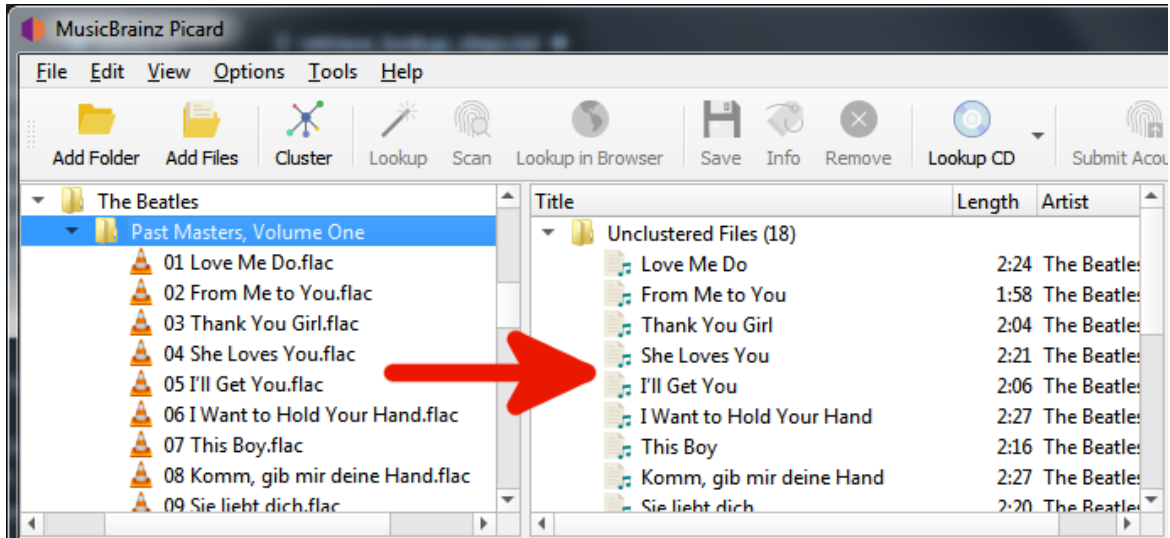
Si aucune des méthodes automatisées n'est disponible ou ne produit pas les résultats souhaités, vous avez la possibilité de récupérer les informations de l'album en demandant à Picard de lancer une recherche sur le site Web MusicBrainz à l'aide de votre navigateur Web. Il existe deux méthodes pour lancer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

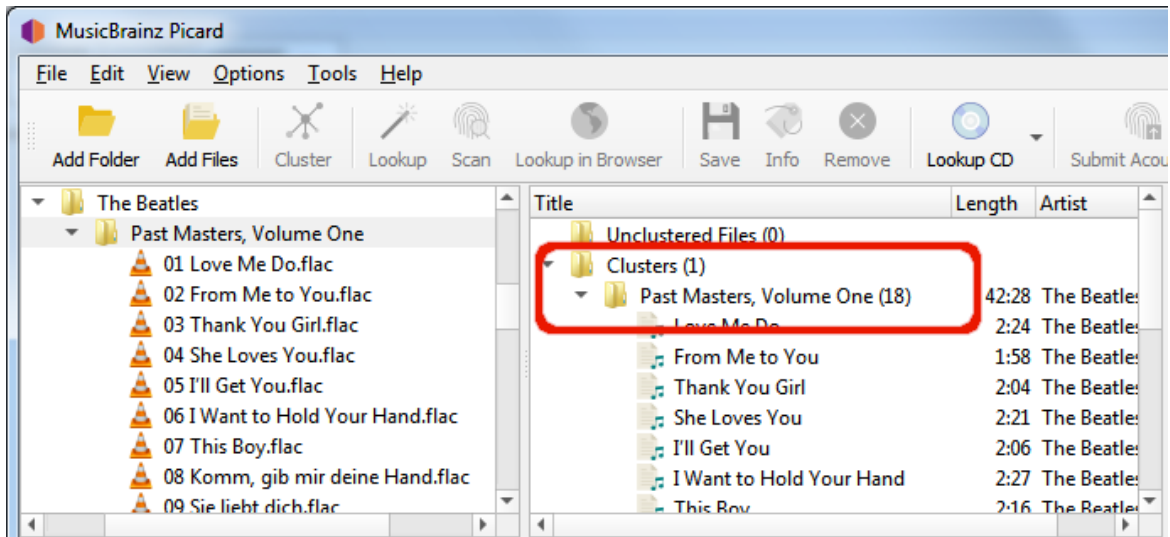
1. Ajoutez vos fichiers en utilisant « *Fichiers → Ajouter des fichiers...* » ou « *Fichiers → Ajouter un dossier...* ». Pour faciliter l'utilisation, il est recommandé d'utiliser le navigateur de fichiers interne pour gérer les interactions du système de fichiers. Ceci est activé à partir de « *Affichage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



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.

MusicBrainz rdsuift - My Data - Search Artist Q

About Us - Products - Search - Editing - Documentation - English - 985

### Tag Lookup Results

Previous 1 2 3 4 5 93 Next

Found 2,321 results

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

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.

All You Need Is Love The Beatles (unknown) 2 US 1967-07-17 eng / Latn Single Official Tagger

Let It Be The Beatles 7" Vinyl 2 US 1970-03-11 Apple Records 2764 [none] eng / Latn Single Official Tagger

Free as a Bird The Beatles (unknown) 2 GB 1995-12-12 eng / Latn Single Official Tagger

Nothing Is Real: The Making of "Strawberry Fields Forever" The Beatles (unknown) 14 DE 1992 eng / Latn Other Bootleg Tagger

Previous 1 2 3 4 5 93 Next

Artist: The Beatles

Release: Past Masters, Volume One

Track number:

Track:

Duration:

Filename:

Search

Donate | Wiki | Forums | IRC | Bug Tracker | Blog | Twitter | Use beta site  
Running: production (8ad15d7222)

Brought to you by MetaBrainz Foundation and our sponsors and supporters. Cover Art provided by the Cover Art Archive.

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

MusicBrainz rdsuift - My Data - Search Artist Q

About Us - Products - Search - Editing - Documentation - English - 985

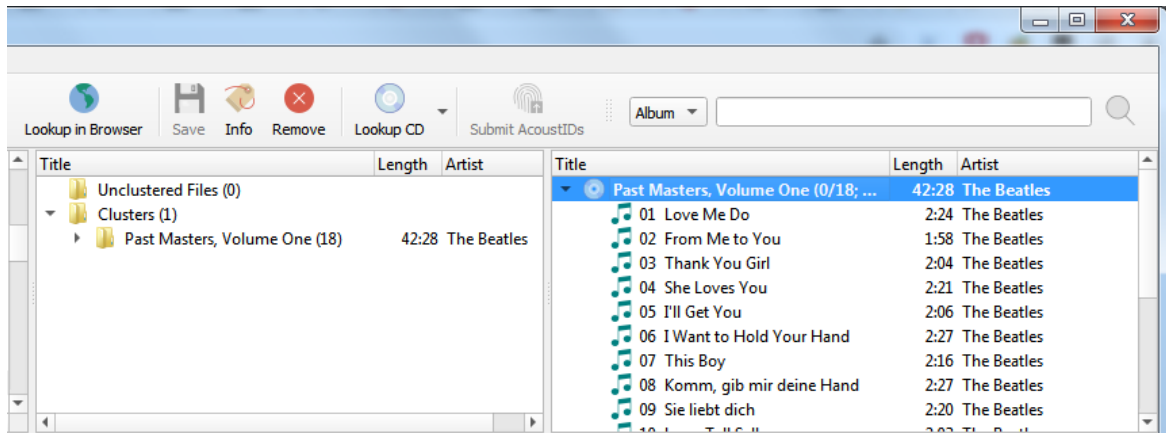
### Tag Lookup Results

Previous 1 2 3 4 5 93 Next

Found 2,321 results

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

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.

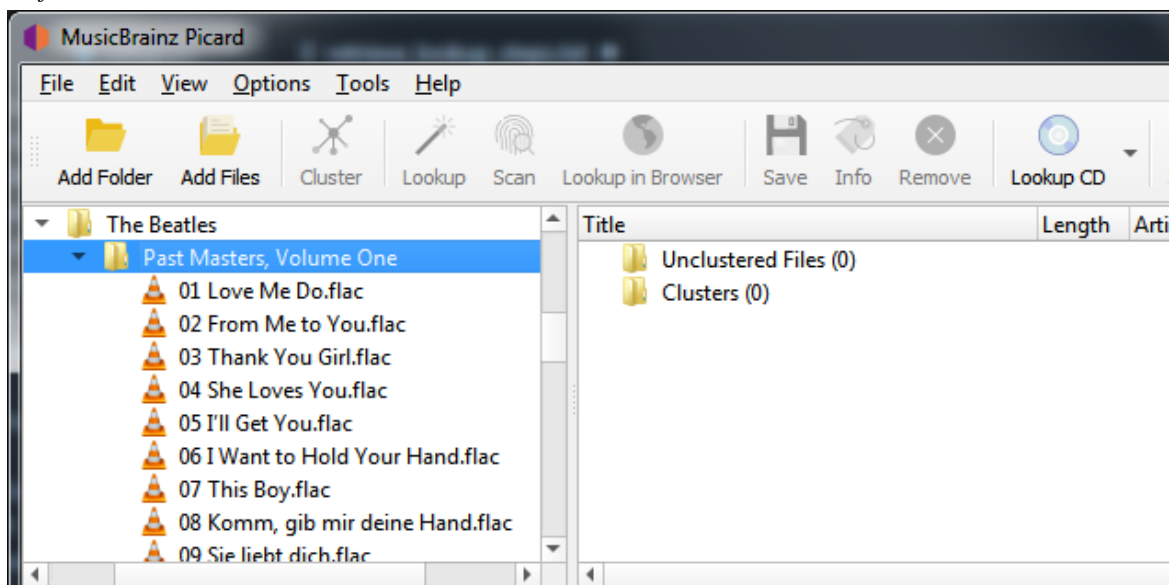


### 8.1.5 Recherche manuelle

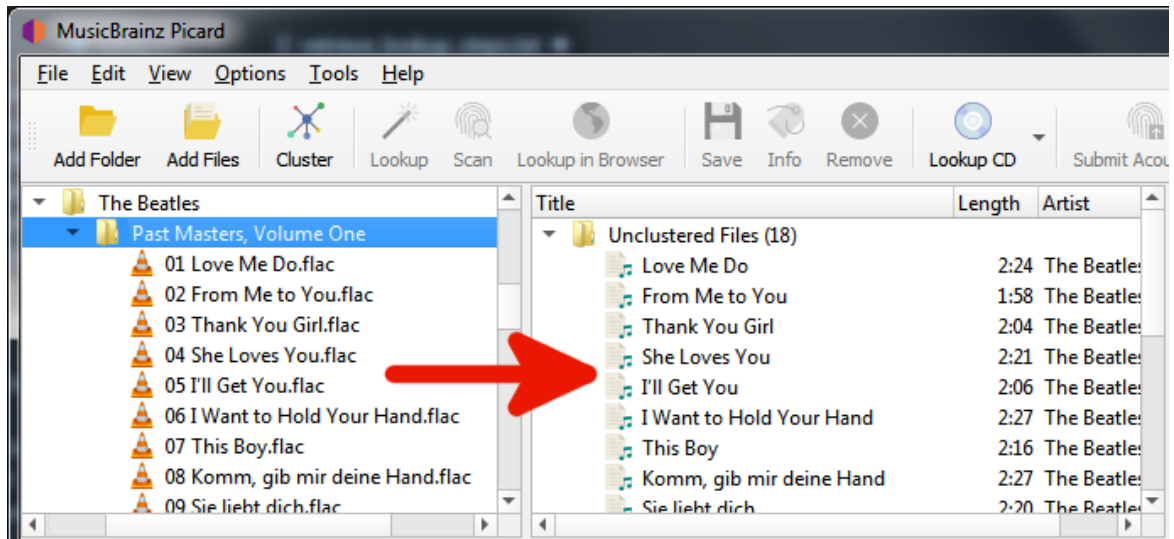
La deuxième méthode de recherche par navigateur utilise des informations saisies manuellement comme critère de recherche.

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

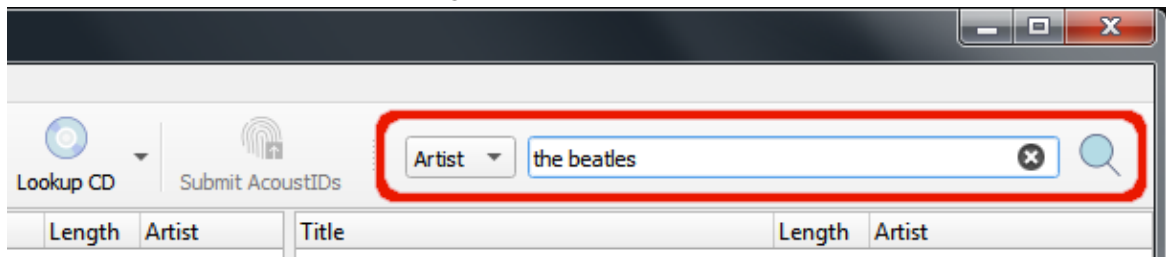
1. Ajoutez vos fichiers en utilisant « *Fichiers → Ajouter des fichiers...* » ou « *Fichiers → Ajouter un dossier...* ». Pour faciliter l'utilisation, il est recommandé d'utiliser le navigateur de fichiers interne pour gérer les interactions du système de fichiers. Ceci est activé à partir de « *Affichage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



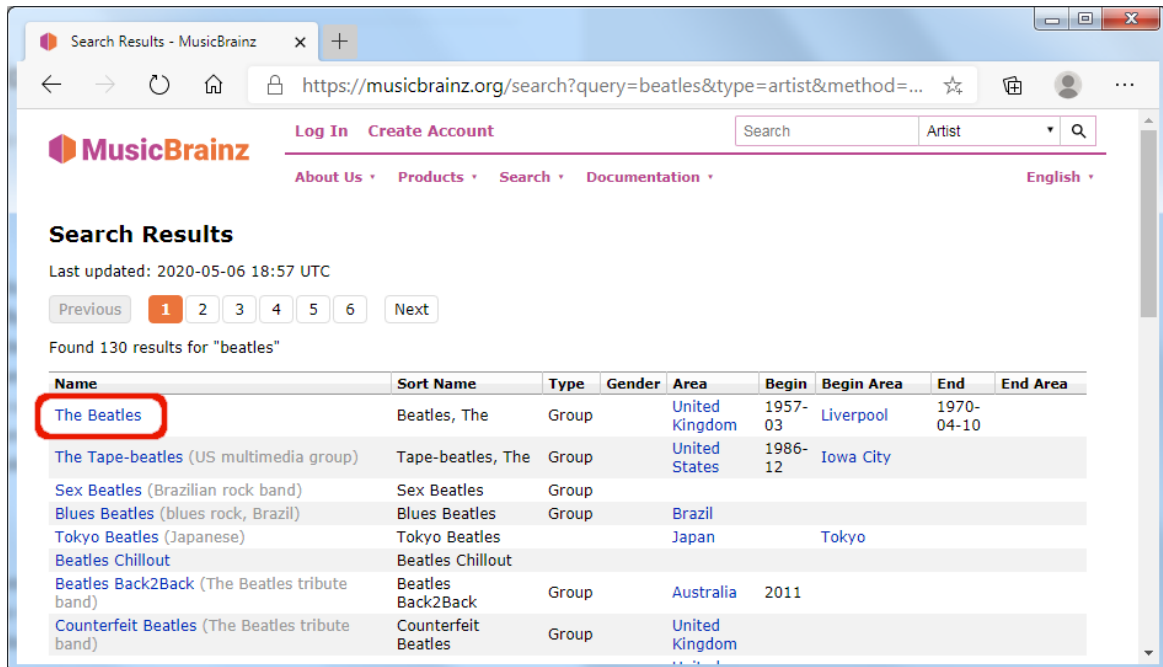
2. Faites glisser le répertoire ou les fichiers sélectionnés vers le dossier « Fichiers non groupés » et attendez que Picard traite les fichiers - les noms passeront du gris au noir.




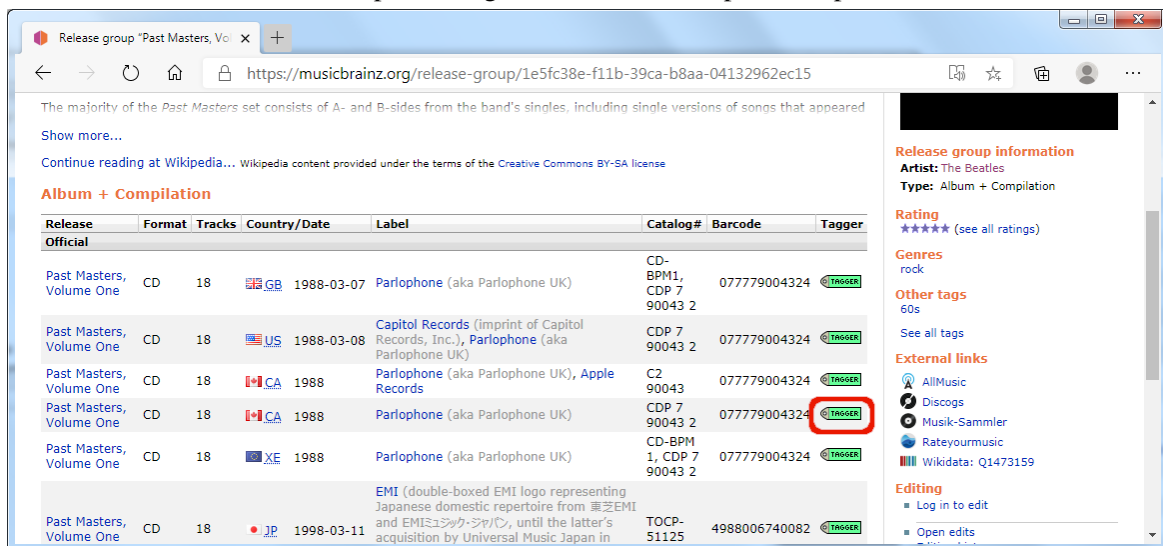
3. Entrez vos informations de recherche dans la zone de recherche et sélectionnez le type d'enregistrements à rechercher, puis cliquez sur le symbole de la loupe pour lancer la recherche. Cela ouvrira le [site Web MusicBrainz](#) dans votre navigateur.



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

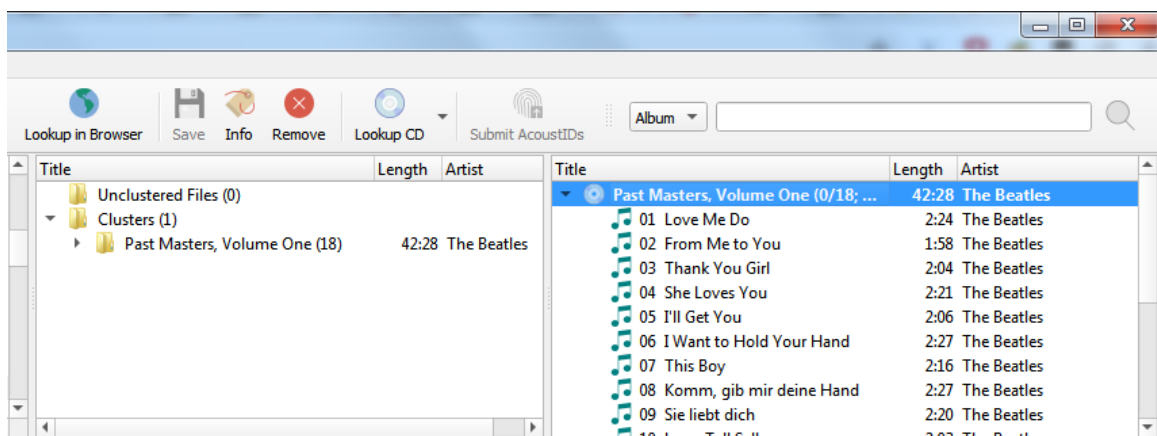


5. Utilisez la flèche verte  pour charger les informations pour une publication dans Picard.



6. 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.



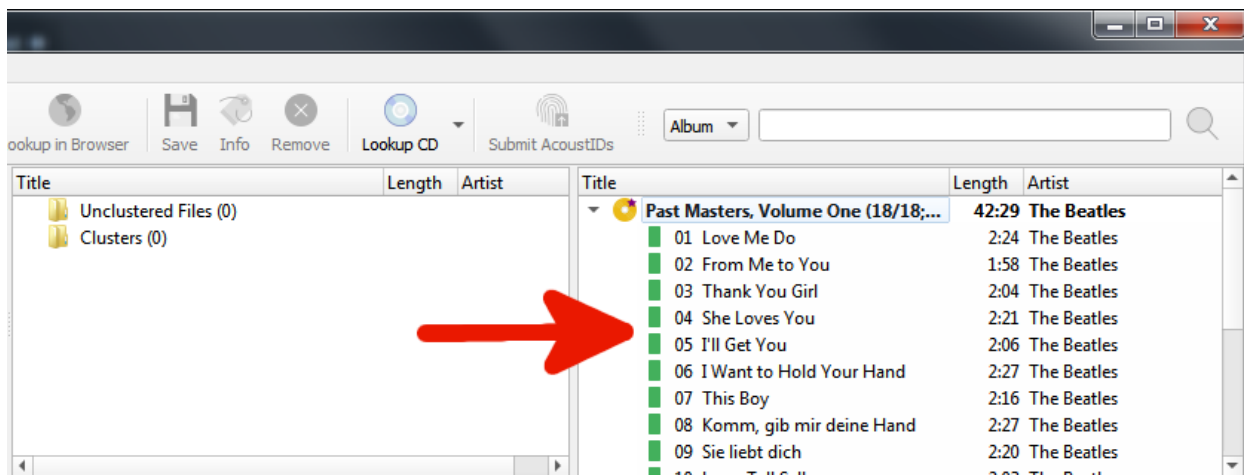


**Note :** Si vous entrez un lien vers l'entrée souhaitée (par exemple : <https://musicbrainz.org/release/9383a6f5-9607-4a36-9c68-8663aad3592b>) dans le champ de recherche en Picard, l'entrée sera chargée directement sans ouvrir une fenêtre de navigateur.

## 8.2 Faire correspondre les fichiers aux pistes

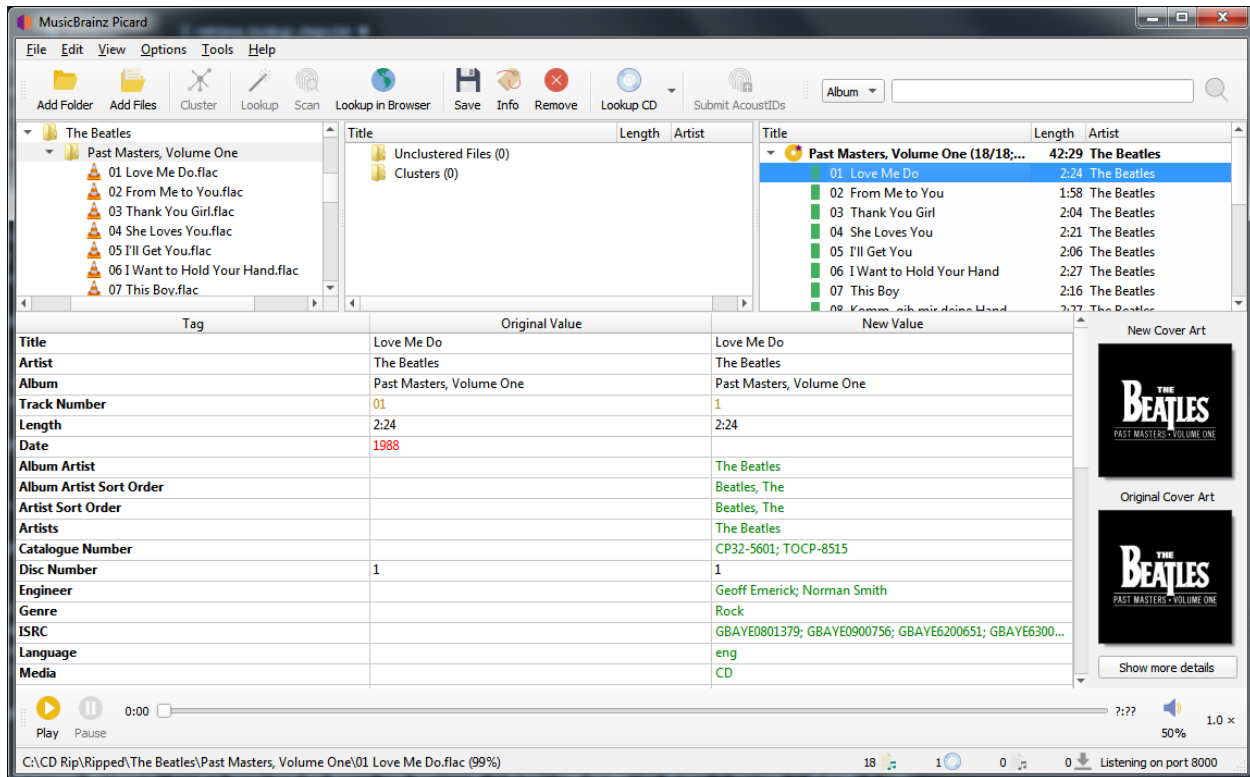
Cette étape est celle où les fichiers individuels sont mis en correspondance avec des pistes spécifiques dans les informations extraites de la base de données MusicBrainz.

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.





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.



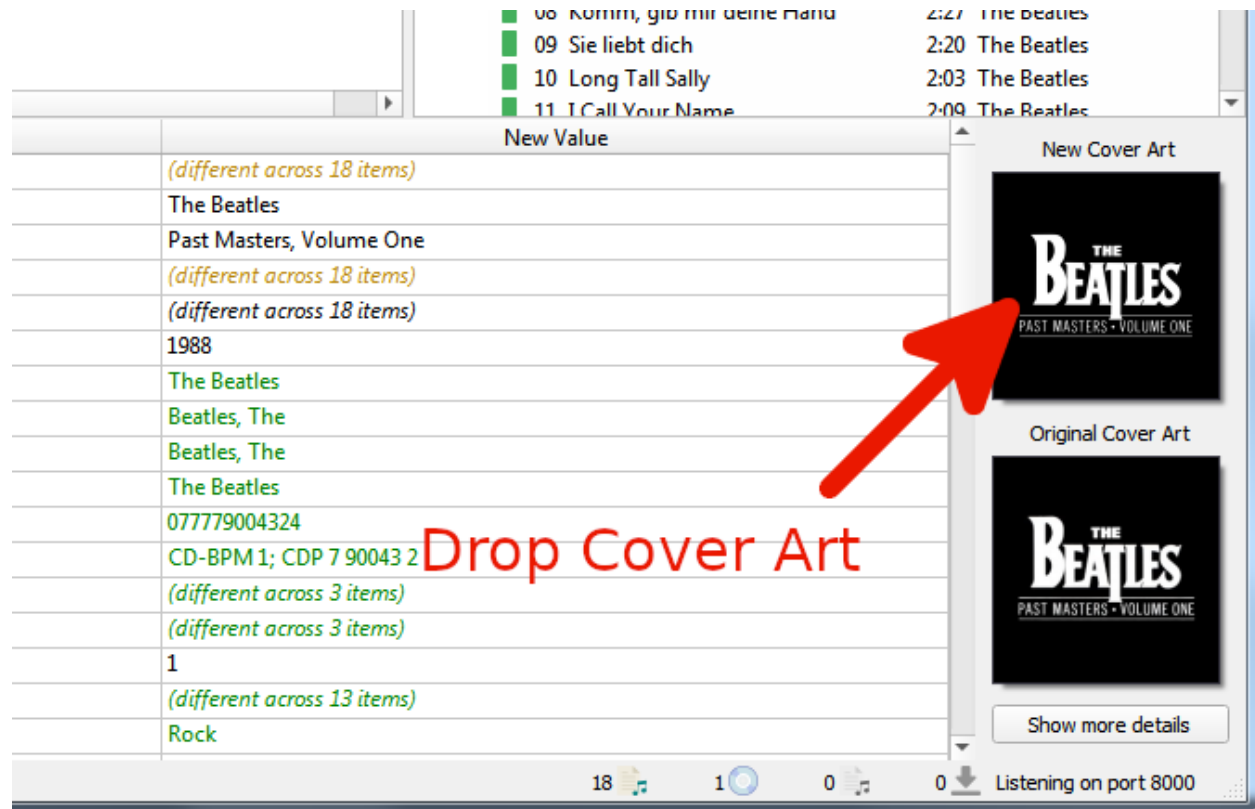
Cliquez avec le bouton droit de la souris sur un élément de la liste des pistes pour afficher un menu de commandes, notamment « Info », « Ouvrir dans le lecteur », « Ouvrir le dossier contenant », « Rechercher des pistes similaires », « Looop dans le navigateur », « Générer des empreintes AcoustID », « 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.

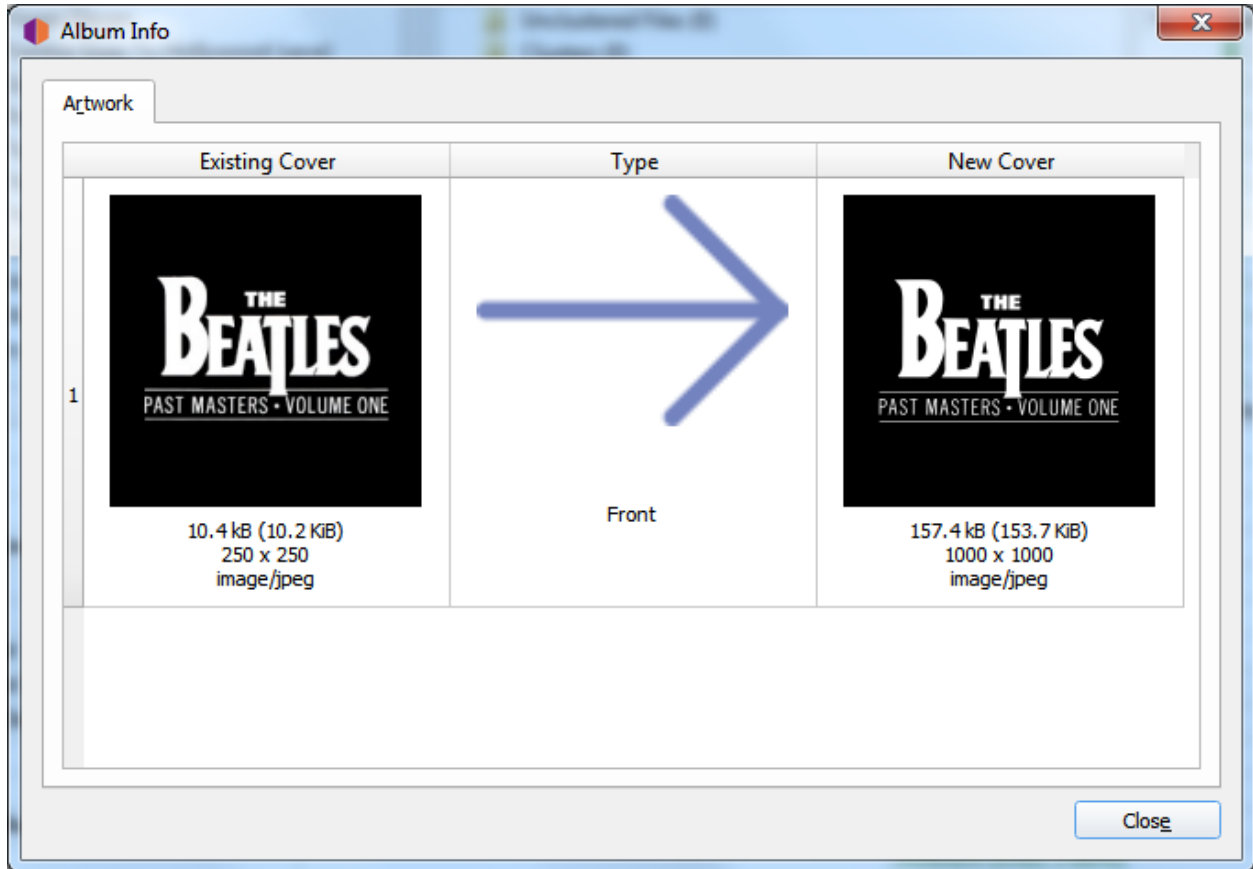
## 8.3 Définition de la pochette

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.



Un clic droit sur les images fait apparaître un menu d'options comprenant « Afficher plus de détails », « Conserver la pochette d'origine » et des options pour le 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.

## 8.4 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 les dossiers. Une coche verte signifie que le fichier a été enregistré à son emplacement approprié.



Title	Length	Artist
▼ ● <b>Past Masters, Volume One (18/18;...</b>	<b>42:28</b>	<b>The Beatles</b>
✓ 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 Should Have Known	2:02	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.

---

### 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.

---

### 9.1 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 *Faire correspondre les 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.

---

## 9.2 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 cluster d'albums à l'aide de la commande « *Outils* → *Cluster* ». Sélectionnez le cluster dans le volet de gauche et lancez la recherche à l'aide de « *Commande 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 *Faire correspondre les 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.

---

## 9.3 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 fonction de leurs métadonnées existantes. Sélectionnez les groupes souhaités dans la partie gauche et lancez la recherche à l'aide de la commande « *Outils* → *Recherche* ». Voir 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 *Faire correspondre les 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.

---

## 9.4 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 volet de gauche et scannez-les à l'aide de la commande « *Outils* → *Analyser* ». Picard tentera de calculer l'empreinte digitale AcoustID pour chacun des fichiers, puis récupérera les versions 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 *Faire correspondre les 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.

## CHAPITRE 10

---

### 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 :

#### 10.1 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.

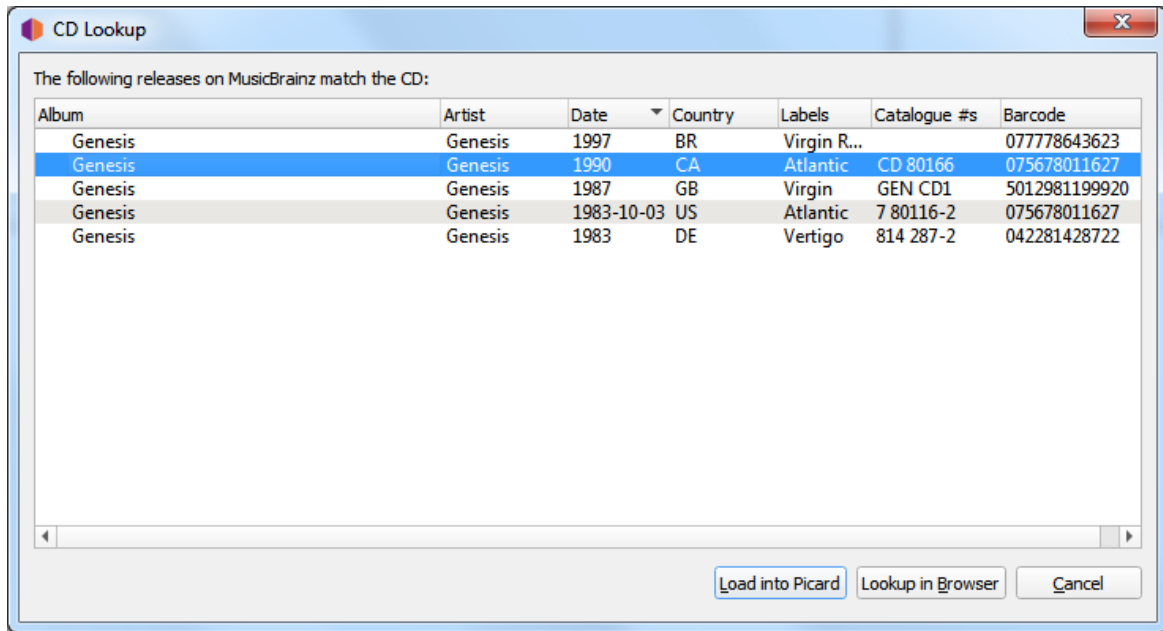
---

**Note :** Please do not add DiscIDs from CDs that are burned at home.

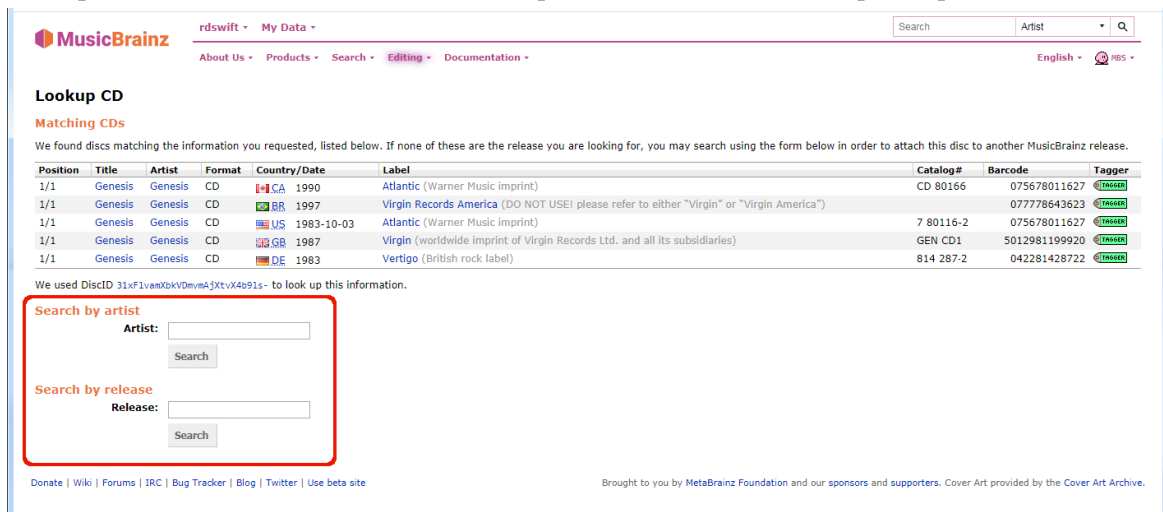
---

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.



2. 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 « *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.



**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 CD TOC » sous la liste des versions. Cela préparera une édition pour joindre l'ID du disque à la version. Ajoutez une note de modification appropriée et soumettez la modification.

Release Group: Splendid Genesis					
Splendid Genesis		Abstract Reason	XW	2015-01-24	[none]
<input type="radio"/> Digital Media 1 (show tracklist)					
Release Group: Genesis					
Genesis		Genesis	GB	1983	Charisma GENMC 1 [none]
<input type="radio"/> Cassette 1 (show tracklist)					
Genesis		Genesis	DE	1983	Vertigo (British rock label) 814 287-2 042281428722
<input type="radio"/> CD 1 (show tracklist)					
Genesis		Genesis	US	1983-10-03	Atlantic (Warner Music imprint)
<input type="radio"/> Digital Media 1 (show tracklist)					
Genesis		Genesis	XE	-	Charisma, Virgin (worldwide imprint of Virgin Records Ltd. and all its subsidiaries) 7 86436 2, GEN CD1 0077778643623
<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 075678011627
<input type="radio"/> CD 1 (show tracklist)					
Genesis (made in Japan)		Genesis	XE	-	Atlantic (Warner Music imprint) 7 80116-2, 814287-2 [none]
<input type="radio"/> 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](#)

## 10.2 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.

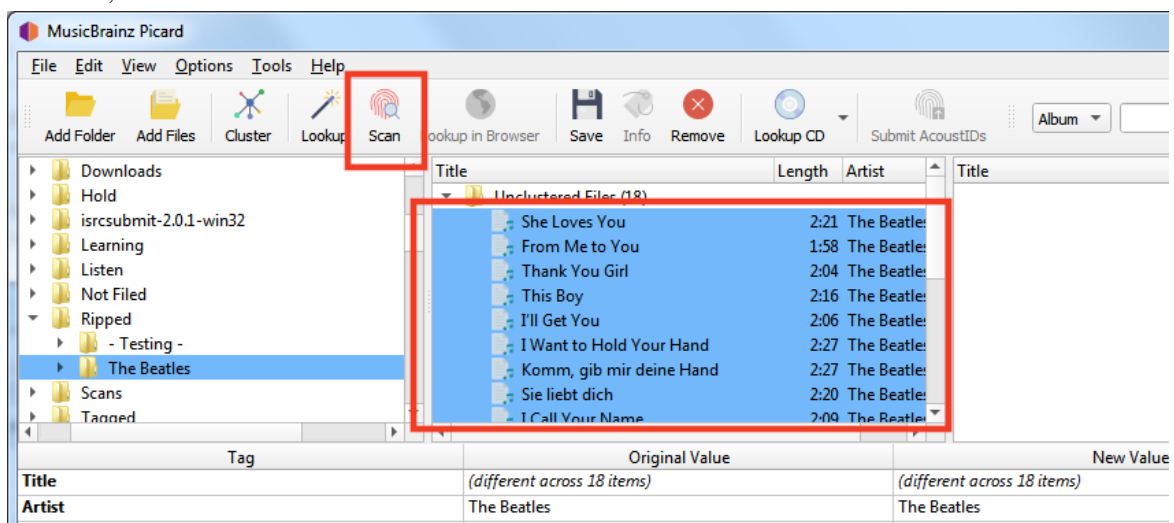
**Note :** Lorsque vous utilisez Picard pour soumettre des empreintes acoustiques, il est recommandé d'activer la colonne Empreintes digitales dans la vue du tableau dans le volet de droite. Pour ce faire, cliquez avec le

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).

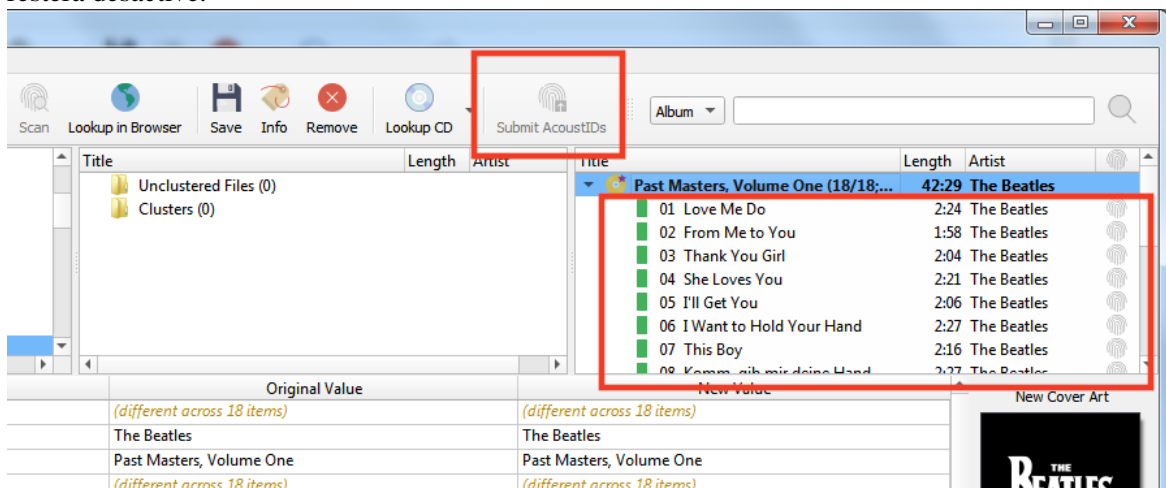
Il existe deux méthodes pour soumettre des empreintes acoustiques, en fonction du flux de travail que vous utilisez pour identifier les versions que vous marquez. Les étapes à suivre pour soumettre des empreintes acoustiques pour chacun des deux flux de travail sont les suivantes :

### 10.2.1 Envoi lors de l'utilisation de Scan pour identifier la version

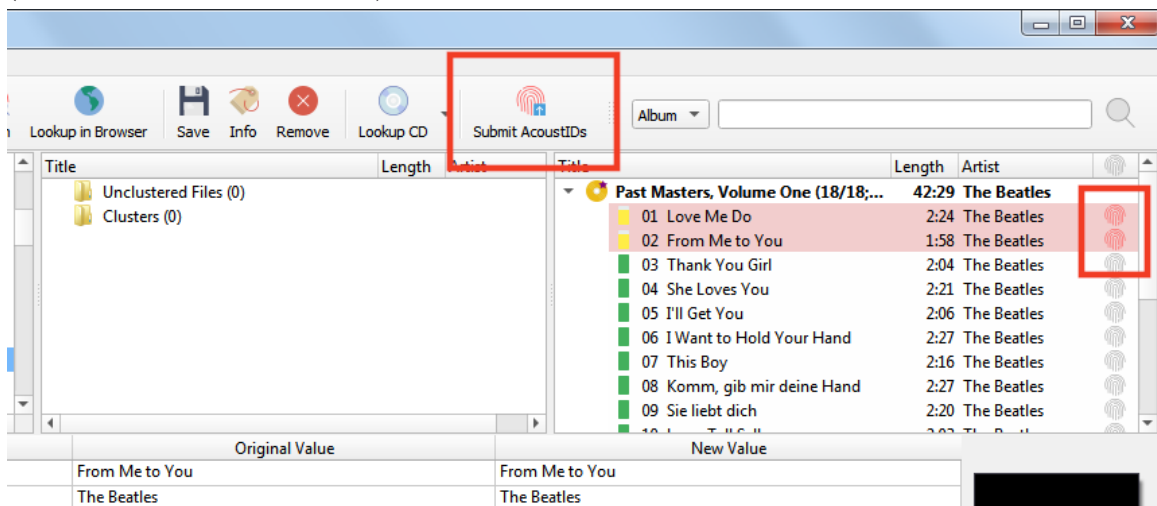
1. Chargez les fichiers dans le volet de clustering. Sélectionnez les fichiers et cliquez sur le bouton « Scan », ou sélectionnez « Outils → Scan ».



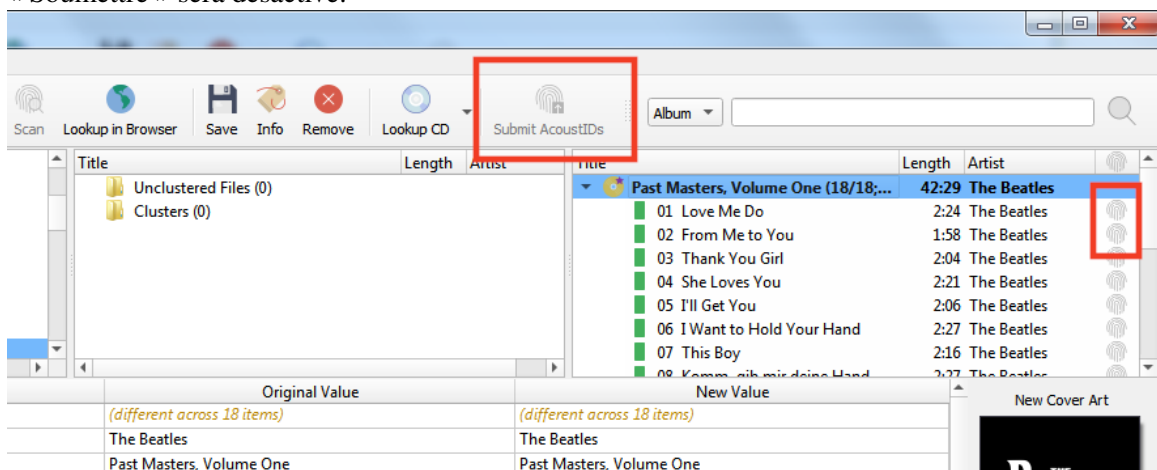
2. Si les fichiers correspondent à une piste et sont déplacés vers le volet de droite, ils existent déjà dans la base de données AcoustID et n'ont pas besoin d'être soumis à nouveau. Le bouton « Soumettre » restera désactivé.



- 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.

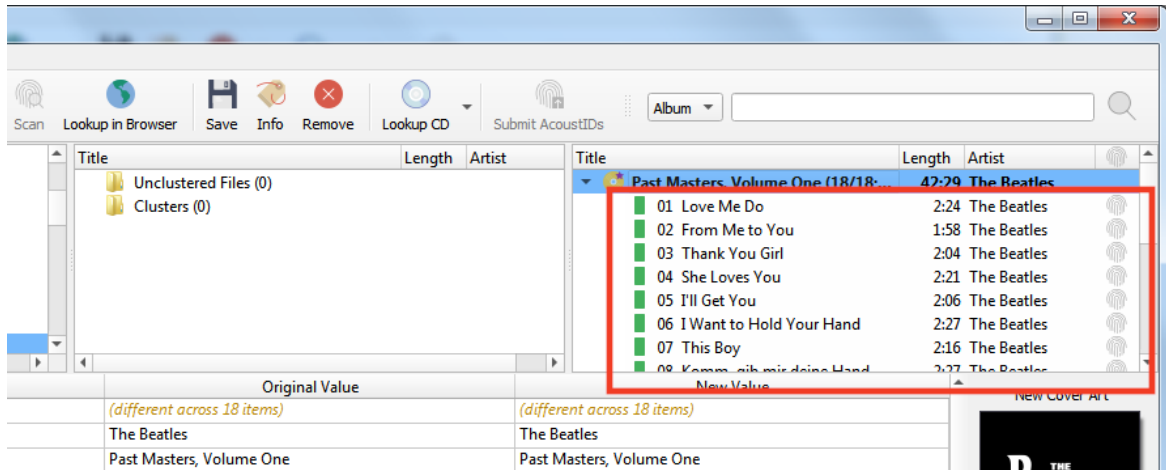


- Cliquer sur le bouton « Soumettre » ne soumettra que les empreintes digitales des fichiers identifiés à l'étape 3. L'icône AcoustID des pistes deviendra grise (c'est-à-dire : statut soumis) et le bouton « Soumettre » sera désactivé.

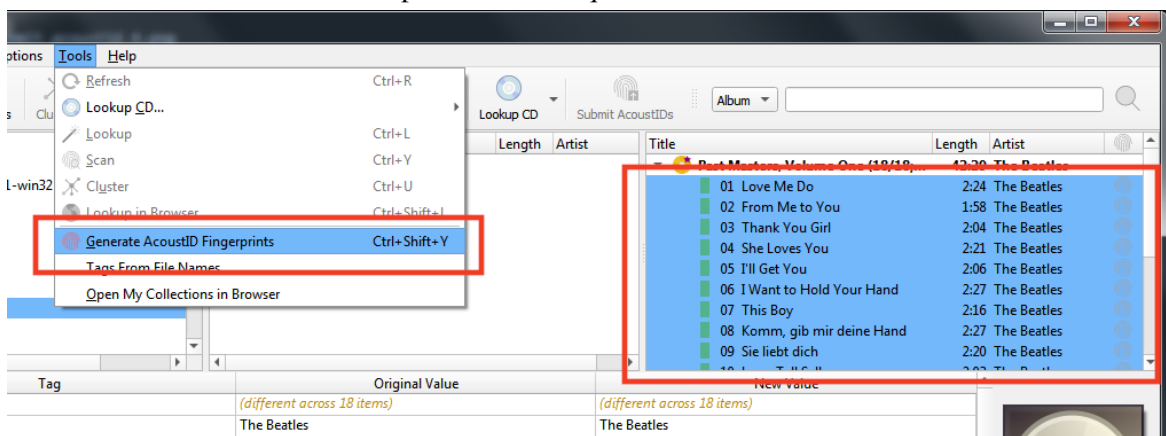


## 10.2.2 Soumission lorsque vous n'utilisez pas Scan pour identifier la version

1. Assurez-vous que les fichiers correspondent correctement aux pistes d'une version dans le volet de droite.

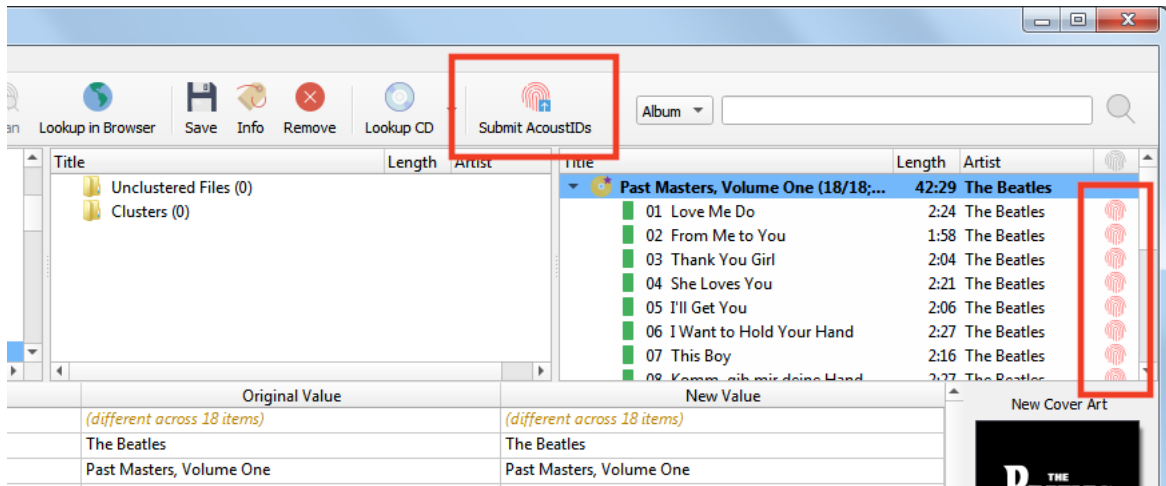


2. Sélectionnez les fichiers dans le volet de droite et sélectionnez « Outils → Générer des empreintes AcoustiD ». Cela calculera les empreintes acoustiques des fichiers sélectionnés.

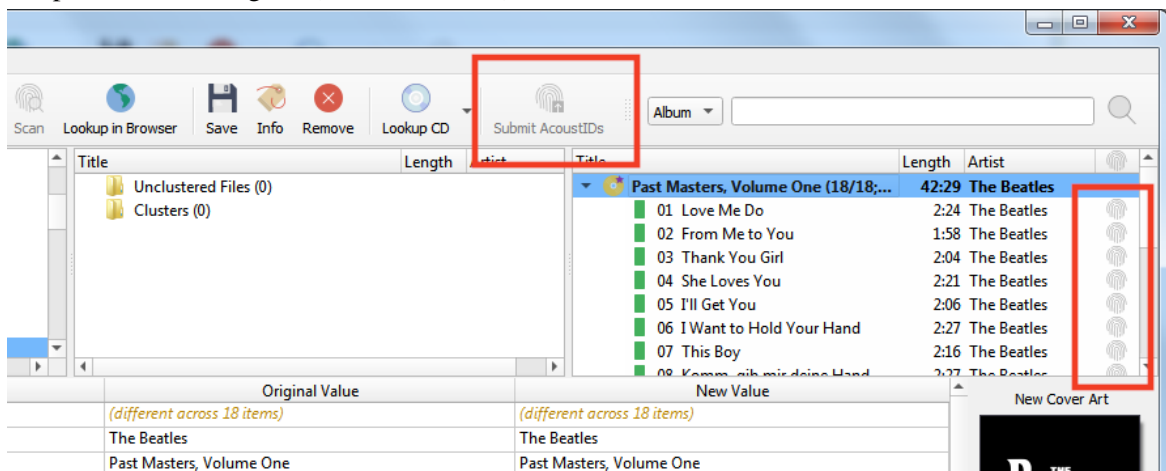


**Note :** Le bouton d'action « Générer des empreintes digitales AcoustiD » peut être ajouté à la barre de boutons en modifiant les paramètres dans les options de l'interface utilisateur.

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.





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...* ».

## 11.1 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.

## 11.2 Les scripts

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...* ». Tous les scripts sont écrits en utilisant le [langage de script Picard](#). 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](#).

### 11.2.1 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.

---

### 11.2.2 Balisage des scripts

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.

## 11.3 Commande en 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.

### 11.3.1 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

### 11.3.2 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.

---

### 11.3.3 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.

### 11.3.4 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.

### 11.3.5 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.

### 11.3.6 Suppression d'albums

Après qu'un album a été supprimé de Picard, les plugins enregistrés avec `album_post_removal_processor()` sont exécutés.

### 11.3.7 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.

### 12.1 Dépannage général

#### 12.1.1 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.

#### 12.1.2 Signaler un bug

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 ?

### 12.1.3 Obtenir 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.

## 12.2 There is no coverart

Il y a deux problèmes différents qui relèvent souvent de ce sujet :

### 12.2.1 Picard ne trouve et ne télécharge aucune pochette

#### Aucun fournisseur de pochettes n'a été activé dans les paramètres de configuration

Confirm that the « Options → Options... → Cover Art » settings have at least one cover art provider enabled. Please see the *Fournisseurs d'art de couverture* section for more information.

#### Aucune pochette n'est disponible auprès des fournisseurs sélectionnés

It's possible that the selected release does not have any cover art available from the enabled cover art providers. If a cover art image is displayed for the release on the MusicBrainz website, it is possible that the image for the release group is being displayed, or it is being provided through a third-party provider agreement. Sometimes this can be addressed by enabling the « CAA Release Group » and « Whitelist » provider options.

#### 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

If the cover art was recently added, the edit adding the image may not have been accepted and applied yet. You can have Picard use the cover art from pending edits by disabling the « Download only approved images » option in the Cover Art Archives subsection of the « Options → Options... → Cover Art » settings. Please see the *Archives d'art de couverture* section for more information.

## 12.2.2 La couverture enregistrée avec les fichiers ne s’affiche pas

### Le lecteur ne prend pas en charge la pochette intégrée

Vérifiez que votre lecteur prend en charge les images de couverture intégrées. Ce soutien n’est pas universel parmi tous les acteurs. Certains lecteurs prennent en charge les images intégrées, certains prennent en charge les images stockées sous forme de fichiers dans le répertoire (par exemple : « cover.jpg » ou « folder.jpg »), et certains prennent en charge les deux. Picard vous permet de spécifier comment les images de la pochette doivent être enregistrées. Veuillez consulter la section [Emplacement](#) des options de 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.

## 12.3 Les balises ne sont ni mises à jour ni 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 [Les 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 et 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

Vérifiez que le type de fichier que vous écrivez prend 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 [../appendices/tag\\_mapping](#) pour plus de détails sur les balises prises en charge par différents formats de fichiers.



## 12.4 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 *Faire correspondre les 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.

## 12.5 Picard vient d'arrêter 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 plug-ins qui effectuent des appels de demande d'informations supplémentaires à l'API du serveur MusicBrains.

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.

---

## 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.

### 13.1 Utiliser Picard

#### Comment marquer des fichiers avec Picard ?

Une section distincte explique le processus de marquage. Veuillez consulter *Utiliser Picard* pour plus de détails.

#### L'icône verte « Tagger » a disparu de MusicBrainz.org, comment puis-je la 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.

#### J'essaye de charger une sortie 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 supprimée. 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 des preuves de la tracklist et autant d'informations que possible pour prouver qu'il est authentique et qu'il peut être accepté à nouveau.

### **J'utilise macOS, où sont mes dossiers ou lecteurs réseau ?**

Ceux-ci 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 souhaitez les voir dans le volet du navigateur de fichiers, faites un clic droit dans le volet et sélectionnez « afficher les fichiers cachés ». Ils doivent alors être visibles dans le dossier /Volumes.

## **13.2 Formats de fichier**

### **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.

---

### 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.

### Quelles balises Picard peut-il écrire dans mes fichiers ?

Voir la section *Balises et 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.

### Comment modifier plusieurs balises à la fois ? Pourquoi n'est-ce pas plus facile de le 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.

### 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.

### **J'utilise Fedora. Pourquoi la prise d'empreintes acoustiques ne fonctionne-t-elle pas ?**

La prise d'empreintes acoustiques en Picard utilise un outil appelé `fpcalc`, qui n'est pas disponible dans Fedora. Vous pouvez l'obtenir en installant le pack `chromaprint-tools` à partir du référentiel [RPM Fusion](#). Cette fonctionnalité n'est pas contenue dans le paquet principal Fedora `picard` car elle nécessite le paquet `ffmpeg` qui [ne peut pas être distribué par Fedora](#). Après avoir activé le référentiel [RPM Fusion](#) « sans `rpmfusion` », installez le package en utilisant (en tant que root) :

```
yum install chromaprint-tools
```

## 13.3 Configuration

### **Où est enregistrée la configuration Picard ?**

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`

### **J'ai tagué un fichier en Picard, mais iTunes ne voit pas les tags !**

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](#).

Additionally, iTunes has a known bug in its ID3v2.4 implementation, which makes it unable to read such tags if they also contain embedded cover art. As a work-around, you can configure Picard to write ID3v2.3 tags.

### Mes balises sont tronquées 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.

### Comment indiquer 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.

## 14.1 Annexe 1 : API des plugins

### 14.1.1 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'exécuter 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.

### 14.1.2 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):
    metadata["title"] = re.sub(r"\(feat. [^)]*\)", "", metadata["title"])

register_track_metadata_processor(remove_featartists)

```

### 14.1.3 Hooks 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

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.

## 14.1.4 Formats de fichier

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 = "...

from picard.file import File
from picard.formats import register_format
from picard.metadata import Metadata

class MyFile(File):
    EXTENSIONS = [".foo"]
    NAME = "Foo Audio"

```

(suite sur la page suivante)

(suite de la page précédente)

```
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)
```

### 14.1.5 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**.

### 14.1.6 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* ».

## 14.2 Annexe 2 : Mappage des balises

Ce qui suit est un mappage entre les noms de balises internes de Picard et ceux utilisés par divers formats de balisage.

### 14.2.1 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

### 14.2.2 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

### 14.2.3 Album

Nom interne	album
ID3v2	TALB
Vorbis	ALBUM
APEv2	Album
iTunes MP4	©alb
ASF/Windows Media	WM/AlbumTitle

### 14.2.4 Artiste de l'album

Nom interne	albumartist
ID3v2	TPE2
Vorbis	ALBUMARTIST
APEv2	Album Artist
iTunes MP4	aART
ASF/Windows Media	WM/AlbumArtist

### 14.2.5 Ordre de tri de l'artiste de l'album

Nom interne	albumartistsort
ID3v2	TSO2 (Picard>=1.2) TXXX:ALBUMARTISTSORT (Picard<=1.1)
Vorbis	ALBUMARTISTSORT
APEv2	ALBUMARTISTSORT
iTunes MP4	soaa
ASF/Windows Media	WM/AlbumArtistSortOrder

### 14.2.6 Ordre de tri de l'album

Nom interne	albumsort
ID3v2	TSOA
Vorbis	ALBUMSORT
APEv2	ALBUMSORT
iTunes MP4	soal
ASF/Windows Media	WM/AlbumSortOrder

Voir le point 4 dans les notes à la fin de cette section.

### 14.2.7 Arrangeur

Nom interne	arranger
ID3v2	TIPL:arranger (id3v24) IPLS:arranger (id3v23)
Vorbis	ARRANGER
APEv2	Arranger
iTunes MP4	n / a
ASF/Windows Media	n / a

### 14.2.8 Artiste

Nom interne	artist
ID3v2	TPE1
Vorbis	ARTIST
APEv2	Artist
iTunes MP4	©ART
ASF/Windows Media	Author

### 14.2.9 Ordre de tri des artistes

Nom interne	artistsort
ID3v2	TSOP
Vorbis	ARTISTSORT
APEv2	ARTISTSORT
iTunes MP4	soar
ASF/Windows Media	WM/ArtistSortOrder

### 14.2.10 Artistes

Nom interne	artists
ID3v2	TXXX:Artists
Vorbis	ARTISTS
APEv2	Artists
iTunes MP4	----:com.apple.iTunes:ARTISTS
ASF/Windows Media	WM/ARTISTS

### 14.2.11 ASIN

Nom interne	asin
ID3v2	TXXX:ASIN
Vorbis	ASIN
APEv2	ASIN
iTunes MP4	----:com.apple.iTunes:ASIN
ASF/Windows Media	ASIN

### 14.2.12 Code à barres

Nom interne	barcode
ID3v2	TXXX:BARCODE
Vorbis	BARCODE
APEv2	Barcode
iTunes MP4	----:com.apple.iTunes:BARCODE
ASF/Windows Media	WM/Barcode



### 14.2.13 BPM

Nom interne	bpm
ID3v2	TBPM
Vorbis	BPM
APEv2	BPM
iTunes MP4	tempo
ASF/Windows Media	WM/BeatsPerMinute

Voir le point 4 dans les notes à la fin de cette section.

### 14.2.14 Numéro de catalogue

Nom interne	catalognumber
ID3v2	TXXX:CATALOGNUMBER
Vorbis	CATALOGNUMBER
APEv2	CatalogNumber
iTunes MP4	----:com.apple.iTunes:CATALOGNUMBER
ASF/Windows Media	WM/CatalogNo

### 14.2.15 Commentaire

Nom interne	comment:description
ID3v2	COMM:description
Vorbis	COMMENT
APEv2	Comment
iTunes MP4	@cmt
ASF/Windows Media	Description

Voir le point 4 dans les notes à la fin de cette section.

### 14.2.16 Compilation (iTunes)

Nom interne	compilation
ID3v2	TCMP
Vorbis	COMPILATION
APEv2	Compilation
iTunes MP4	cpil
ASF/Windows Media	WM/IsCompilation

Voir le point 5 dans les notes à la fin de cette section.

### 14.2.17 Compositeur

Nom interne	composer
ID3v2	TCOM
Vorbis	COMPOSER
APEv2	Composer
iTunes MP4	©wrt
ASF/Windows Media	WM/Composer

### 14.2.18 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)

### 14.2.19 Chef d'orchestre

Nom interne	conductor
ID3v2	TPE3
Vorbis	CONDUCTOR
APEv2	Conductor
iTunes MP4	----:com.apple.iTunes:CONDUCTOR
ASF/Windows Media	WM/Conductor

### 14.2.20 Copyright

Nom interne	copyright
ID3v2	TCOP
Vorbis	COPYRIGHT
APEv2	Copyright
iTunes MP4	cp rt
ASF/Windows Media	Copyright

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.21 Numéro de disque

Nom interne	discnumber
ID3v2	TPOS
Vorbis	DISCNUMBER
APEv2	Disc
iTunes MP4	disk
ASF/Windows Media	WM/PartOfSet

### 14.2.22 Sous-titre du disque

Nom interne	discsubtitle
ID3v2	TSST (id3v24 only)
Vorbis	DISCSUBTITLE
APEv2	DiscSubtitle
iTunes MP4	----:com.apple.iTunes:DISCSUBTITLE
ASF/Windows Media	WM/SetSubTitle

### 14.2.23 Encodé par

Nom interne	encodedby
ID3v2	TENC
Vorbis	ENCODEDBY
APEv2	EncodedBy
iTunes MP4	@too
ASF/Windows Media	WM/EncodedBy

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.24 Paramètres de l'encodeur

Nom interne	encodersettings
ID3v2	TSSE
Vorbis	ENCODERSETTINGS
APEv2	EncoderSettings
iTunes MP4	n / a
ASF/Windows Media	WM/EncodingSettings (Picard>=1.3.1)

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.25 Ingénieur

Nom interne	engineer
ID3v2	TIPL:engineer (id3v24) IPLS:engineer (id3v23)
Vorbis	ENGINEER
APEv2	Engineer
iTunes MP4	----:com.apple.iTunes:ENGINEER
ASF/Windows Media	WM/Engineer

### 14.2.26 Gapless Playback

Nom interne	gapless
ID3v2	n / a
Vorbis	n / a
APEv2	n / a
iTunes MP4	pgap
ASF/Windows Media	n / a

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.27 Genre

Nom interne	genre
ID3v2	TCON
Vorbis	GENRE
APEv2	Genre
iTunes MP4	@gen
ASF/Windows Media	WM/Genre

### 14.2.28 Regroupement

Nom interne	grouping
ID3v2	TIT1 GRP1
Vorbis	GROUPING
APEv2	Grouping
iTunes MP4	@grp
ASF/Windows Media	WM/ContentGroupDescription

Voir les points 3 et 8 dans les notes à la fin de cette section.

### 14.2.29 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

### 14.2.30 ISRC

Nom interne	isrc
ID3v2	TSRC
Vorbis	ISRC
APEv2	ISRC
iTunes MP4	----:com.apple.iTunes:ISRC
ASF/Windows Media	WM/ISRC

### 14.2.31 Langue

Nom interne	language
ID3v2	TLAN
Vorbis	LANGUAGE
APEv2	Language
iTunes MP4	----:com.apple.iTunes:LANGUAGE
ASF/Windows Media	WM/Language

### 14.2.32 License

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

*Voir les points 6 et 7 dans les notes à la fin de cette section.*

### 14.2.33 Parolier

Nom interne	lyricist
ID3v2	TEXT
Vorbis	LYRICIST
APEv2	Lyricist
iTunes MP4	----:com.apple.iTunes:LYRICIST
ASF/Windows Media	WM/Writer

### 14.2.34 Paroles

Nom interne	lyrics:description
ID3v2	USLT:description
Vorbis	LYRICS
APEv2	Lyrics
iTunes MP4	©lyr
ASF/Windows Media	WM/Lyrics

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.35 Médias

Nom interne	media
ID3v2	TMED
Vorbis	MEDIA
APEv2	Media
iTunes MP4	----:com.apple.iTunes:MEDIA
ASF/Windows Media	WM/Media

### 14.2.36 Mix-DJ

Nom interne	djmixer
ID3v2	TIPL:DJ-mix (id3v24) IPLS:DJ-mix (id3v23)
Vorbis	DJMIXER
APEv2	DJMixer
iTunes MP4	----:com.apple.iTunes:DJMIXER
ASF/Windows Media	WM/DJMixer

### 14.2.37 Mélangeur

Nom interne	mixer
ID3v2	TIPL:mix (id3v24) IPLS:mix (id3v23)
Vorbis	MIXER
APEv2	Mixer
iTunes MP4	----:com.apple.iTunes:MIXER
ASF/Windows Media	WM/Mixer

### 14.2.38 Mood

Nom interne	mood
ID3v2	TMOO (id3v24 only)
Vorbis	MOOD
APEv2	Mood
iTunes MP4	----:com.apple.iTunes:MOOD
ASF/Windows Media	WM/Mood

Voir le point 3 dans les notes à la fin de cette section.

### 14.2.39 Mouvement

Nom interne	movement (Picard>=2.1)
ID3v2	MVNM
Vorbis	MOVEMENTNAME
APEv2	MOVEMENTNAME
iTunes MP4	©mvn
ASF/Windows Media	n / a

Voir le point 4 dans les notes à la fin de cette section.

### 14.2.40 Nombre de mouvements

Nom interne	movementtotal (Picard>=2.1)
ID3v2	MVIN
Vorbis	MOVEMENTTOTAL
APEv2	MOVEMENTTOTAL
iTunes MP4	mvc
ASF/Windows Media	n / a

Voir le point 4 dans les notes à la fin de cette section.

### 14.2.41 Numéro de mouvement

Nom interne	movementnumber (Picard>=2.1)
ID3v2	MVIN
Vorbis	MOVEMENT
APEv2	MOVEMENT
iTunes MP4	mvi
ASF/Windows Media	n / a

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.42 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

### 14.2.43 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

### 14.2.44 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)



#### 14.2.45 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)

#### 14.2.46 Identifiant d'enregistrement MusicBrainz

Nom interne	musicbrainz_recordingid
ID3v2	UFID: `//musicbrainz.org <https://musicbrainz.org>`_
Vorbis	MUSICBRAINZ_TRACKID
APEv2	MUSICBRAINZ_TRACKID
iTunes MP4	----:com.apple.iTunes:MusicBrainz Track Id
ASF/Windows Media	MusicBrainz/Track Id

#### 14.2.47 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

#### 14.2.48 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

### 14.2.49 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

### 14.2.50 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

### 14.2.51 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

### 14.2.52 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

### 14.2.53 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

### 14.2.54 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

### 14.2.55 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)

### 14.2.56 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)

## 14.2.57 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

## 14.2.58 Date de sortie d'origine

Nom interne	originaldate
ID3v2	TDOR (id3v24) TORY (id3v23)
Vorbis	ORIGINALDATE
APEv2	n / a
iTunes MP4	n / a
ASF/Windows Media	WM/OriginalReleaseTime (Picard>=1.3.1) WM/ OriginalReleaseYear (Picard<=1.3.0)

Voir le point 1 dans les notes à la fin de cette section.

## 14.2.59 Année de sortie d'origine

Nom interne	originalyear
ID3v2	n / a
Vorbis	ORIGINALYEAR
APEv2	ORIGINALYEAR
iTunes MP4	n / a
ASF/Windows Media	WM/OriginalReleaseYear (Picard>=1.3.1)

Voir le point 1 dans les notes à la fin de cette section.

## 14.2.60 L'exécutant

Nom interne	performer:instrument
ID3v2	TMCL:instrument (id3v24) IPLS:instrument (id3v23)
Vorbis	PERFORMER={artist} (instrument)
APEv2	Performer={artist} (instrument)
iTunes MP4	n / a
ASF/Windows Media	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.

**14.2.61 Podcast**

Nom interne	podcast
ID3v2	n / a
Vorbis	n / a
APEv2	n / a
iTunes MP4	pcst
ASF/Windows Media	n / a

*Voir le point 4 dans les notes à la fin de cette section.*

**14.2.62 URL du podcast**

Nom interne	podcasturl
ID3v2	n / a
Vorbis	n / a
APEv2	n / a
iTunes MP4	purl
ASF/Windows Media	n / a

*Voir le point 4 dans les notes à la fin de cette section.*

**14.2.63 Producteur**

Nom interne	producer
ID3v2	TIPL:producer (id3v24) IPLS:producer (id3v23)
Vorbis	PRODUCER
APEv2	Producer
iTunes MP4	----:com.apple.iTunes:PRODUCER
ASF/Windows Media	WM/Producer

### 14.2.64 Évaluation

Nom interne	<code>_rating</code>
ID3v2	POPM
Vorbis	<code>RATING:user@email</code>
APEv2	<code>n / a</code>
iTunes MP4	<code>n / a</code>
ASF/Windows Media	<code>WM/SharedUserRating</code>

### 14.2.65 Étiquette d'enregistrement

Nom interne	<code>label</code>
ID3v2	TPUB
Vorbis	LABEL
APEv2	Label
iTunes MP4	<code>----:com.apple.iTunes:LABEL</code>
ASF/Windows Media	<code>WM/Publisher</code>

### 14.2.66 Pays de sortie

Nom interne	<code>releasecountry</code>
ID3v2	<code>TXXX:MusicBrainz Album Release Country</code>
Vorbis	RELEASECOUNTRY
APEv2	RELEASECOUNTRY
iTunes MP4	<code>----:com.apple.iTunes:MusicBrainz Album Release Country</code>
ASF/Windows Media	<code>MusicBrainz/Album Release Country</code>

### 14.2.67 Date de sortie

Nom interne	<code>date</code>
ID3v2	<code>TDRC (id3v24) TYER + TDAT (id3v23)</code>
Vorbis	DATE
APEv2	Year
iTunes MP4	<code>©day</code>
ASF/Windows Media	<code>WM/Year</code>

### 14.2.68 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

### 14.2.69 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

### 14.2.70 Remixeur

Nom interne	remixer
ID3v2	TPE4
Vorbis	REMIXER
APEv2	MixArtist
iTunes MP4	----:com.apple.iTunes:REMIXER
ASF/Windows Media	WM/ModifiedBy

### 14.2.71 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

### 14.2.72 ReplayGain Album Peak

Nom interne	replaygain_album_peak (Picard $\geq$ 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

### 14.2.73 ReplayGain Album Range

Nom interne	replaygain_album_range (Picard $\geq$ 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

### 14.2.74 ReplayGain Reference Loudness

Nom interne	replaygain_reference_loudness (Picard $\geq$ 2.2)
ID3v2	TXXX:REPLAYGAIN_REFERENCE_LOUDNESS
Vorbis	REPLAYGAIN_REFERENCE_LOUDNESS
APEv2	REPLAYGAIN_REFERENCE_LOUDNESS
iTunes MP4	----:com.apple.iTunes:REPLAYGAIN_REFERENCE_LOUDNESS
ASF/Windows Media	REPLAYGAIN_REFERENCE_LOUDNESS

### 14.2.75 ReplayGain Track Gain

Nom interne	replaygain_track_gain (Picard $\geq$ 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



### 14.2.76 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

### 14.2.77 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

### 14.2.78 Script

Nom interne	script
ID3v2	TXXX:SCRIPT
Vorbis	SCRIPT
APEv2	Script
iTunes MP4	----:com.apple.iTunes:SCRIPT
ASF/Windows Media	WM/Script

### 14.2.79 Afficher le nom

Nom interne	show
ID3v2	n / a
Vorbis	n / a
APEv2	n / a
iTunes MP4	tvsh
ASF/Windows Media	n / a

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.80 Afficher l'ordre de tri des noms

Nom interne	showsort
ID3v2	n / a
Vorbis	n / a
APEv2	n / a
iTunes MP4	sosn
ASF/Windows Media	n / a

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.81 Afficher le travail et le mouvement

Nom interne	showmovement (Picard>=2.1)
ID3v2	TXXX:SHOWMOVEMENT
Vorbis	SHOWMOVEMENT
APEv2	SHOWMOVEMENT
iTunes MP4	shwm
ASF/Windows Media	n / a

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.82 Subtitle

Nom interne	subtitle
ID3v2	TIT3
Vorbis	SUBTITLE
APEv2	Subtitle
iTunes MP4	----:com.apple.iTunes:SUBTITLE
ASF/Windows Media	WM/SubTitle

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.83 Nombre total de disques

Nom interne	totaldiscs
ID3v2	TPOS
Vorbis	DISCTOTAL and TOTALDISCS
APEv2	Disc
iTunes MP4	disk
ASF/Windows Media	WM/PartOfSet (Picard>=1.3.1)

### 14.2.84 Total des pistes

Nom interne	totaltracks
ID3v2	TRCK
Vorbis	TRACKTOTAL and TOTALTRACKS
APEv2	Track
iTunes MP4	trkn
ASF/Windows Media	n / a

### 14.2.85 Numéro de piste

Nom interne	tracknumber
ID3v2	TRCK
Vorbis	TRACKNUMBER
APEv2	Track
iTunes MP4	trkn
ASF/Windows Media	WM/TrackNumber

### 14.2.86 Titre de la piste

Nom interne	title
ID3v2	TIT2
Vorbis	TITLE
APEv2	Title
iTunes MP4	©nam
ASF/Windows Media	Title

### 14.2.87 Ordre de tri des titres de piste

Nom interne	titlesort
ID3v2	TSOT
Vorbis	TITLESORT
APEv2	TITLESORT
iTunes MP4	sonm
ASF/Windows Media	WM/TitleSortOrder

*Voir le point 4 dans les notes à la fin de cette section.*

### 14.2.88 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)

### 14.2.89 Work Title

Nom interne	work (Picard>=1.3)
ID3v2	TXXX:WORK TIT1
Vorbis	WORK
APEv2	WORK
iTunes MP4	©wrk (Picard>=2.1)
ASF/Windows Media	WM/Work

Voir le point 8 dans les notes à la fin de cette section.

### 14.2.90 Écrivain

Nom interne	writer
ID3v2	TXXX:Writer (Picard>=1.3)
Vorbis	WRITER
APEv2	Writer
iTunes MP4	n / a
ASF/Windows Media	n / a

Voir le point 2 dans les notes à la fin de cette section.

#### Notes :

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)

## 14.3 Annexe 3 : 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

-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 est le ou les fichiers à charger

## A

- acknowledgements, 2
- acoustic fingerprint, 4, 146
  - submitting, 122, 123, 128
- AcoustID, 4
  - submitting, 128
- albumartist, 4
- api
  - plugins, 148
- artist, 4
- artist credit, 5
- audio player, 145

## B

- browser
  - configuration, 147

## C

- CAA, *voir* cover art archive
- command line
  - options, 177
- configuration
  - aac tag options, 24
  - ac3 tag options, 25
  - action options, 11
  - advanced options, 46
  - before tagging, 21
  - browser, 147
  - cd lookup, 36
  - colors, 43
  - config file location, 146
  - cover art, 28
  - cover art archive, 30
  - cover art location, 28
  - cover art providers, 29

- file naming, 33
- fingerprinting, 35
- general options, 13
- genres, 17
- id3 tag options, 22
- local files, 32
- matching preferences, 49
- metadata options, 15
- network, 47
- plugins, 38
- ratings, 19
- release preferences, 16
- screen setup, 11
- scripting, 45
- tag options, 20
- top tags, 44
- update checking, 14
- user interface, 42
- wave tag options, 26
- context menu actions
  - plugins, 152
- contributing, 2
- cover art
  - configuration, 28
  - location to save, 28
  - setting, 118
- cover art archive, 5
  - configuration, 30

## D

- disc id, 5
  - attaching, 126

## E

- event hooks

plugins, 149

## F

file format  
     plugins, 151  
 file formats, 144  
 file naming  
     configuration, 33  
     scripts, 134  
 files  
     saving, 119  
 fingerprint  
     acoustic, 4, 146  
     submitting, 128  
 flatpak  
     install, 8

## G

glossary, 4

## I

icon  
     tagger, 143  
 icons  
     album, 9  
     release, 9  
     status, 9  
     track, 10  
 install  
     download, 7  
     flatpak, 8  
 itunes, 146

## L

limitations, 3  
 lookup cd, 103  
 lookup files, 106  
 lookup in browser, 110

## M

mapping  
     tags, 154  
 matching files to tracks, 116  
 mbid, *voir* MusicBrainz Identifier  
 MusicBrainz Identifier, 5

## N

non-album track, *voir* standalone recording

## O

option settings, *voir* configuration  
 options  
     command line, 177

## P

plugins, 133  
     api, 148  
     configuration, 38  
     context menu actions, 152  
     event hooks, 149  
     file format, 151  
     installing, 40  
     metadata, 148  
     metadata processors, 149  
     programming, 148  
     scripting functions, 152  
     third-party, 38  
     types, 133  
 processing order, 135  
 programming  
     plugins, 148

## R

rate limiting, 3  
 recording, 5  
     standalone, 6  
 release, 5  
 release group, 6

## S

saving files, 119  
 scan files, 108  
 scripting  
     functions, 66  
 scripting functions  
     assignment, 66  
     conditional, 89  
     information, 100  
     loop, 101  
     mathematical, 86  
     miscellaneous, 102  
     multi-value, 81  
     plugins, 152  
     text, 69  
 scripts, 64, 134  
     file naming, 34, 134  
     syntax, 64

- tagging, [45](#), [135](#)
- tags, [51](#)
- variables, [51](#)
- standalone recording, [6](#)

## T

- tagger icon, [143](#)
- tagging
  - scripts, [135](#)
- tags
  - advanced, [56](#)
  - basic, [52](#)
  - classical, [59](#)
  - editing, [145](#)
  - genre, [57](#)
  - mapping, [154](#)
  - plugins, [60](#)
  - writing, [63](#)
- track, [6](#)
- troubleshooting
  - files not saved, [140](#)
  - general, [138](#)
  - logs, [138](#)
  - no cover art displayed, [139](#)
  - no cover art downloaded, [139](#)
  - program freezes, [141](#)
  - reporting a bug, [138](#)
  - tags not saved, [140](#)

## U

- update checking
  - configuration, [14](#)
- user interface
  - colors, [43](#)
  - configuration, [42](#)
  - main screen, [8](#)

## V

- variables
  - advanced, [59](#)
  - basic, [57](#)

## W

- work, [6](#)
- workflows
  - cd, [121](#)
  - files grouped by album, [122](#)
  - files not grouped, [123](#)

- general, [121](#)
- no metadata, [124](#)