

Manuel de référence

DiscretePhoton H.264 encoder

Manuel de référence

À propos de DiscretePhoton H.264 encoder

DiscretePhoton H.264 encoder Windows version est fourni DirectX Media Object (DMO) en 32-bit ou 64-bit binaire. Il peut être utilisé directement ou dans l'environnement de DirectShow.

DiscretePhoton H.264 encoder supporte jusqu'à 64 threads simultanés. Mais le nombre réel de threads est déterminé par le nombre de CPU-cores (ou hyper-threads) de votre système et la taille de frame. Vous ne pouvez la diminuer.

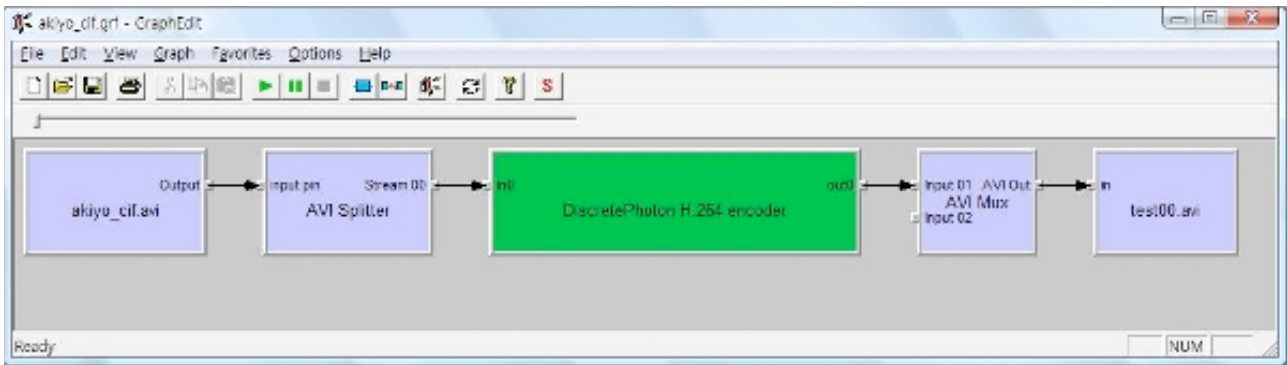
Multi-threading méthode de DiscretePhoton H.264 encoder est basée sur wavefront. Tous les threads de travail sont concentrés sur la frame d'entrée la plus récente, ce qui pourrait entraîner une latence très faible d'encodage sans aucune pénalité sur la qualité finale ou codée bits de taille. En outre, son taux de CBR rate-control est à l'écoute pour le scénario de faible latence. Ainsi, DiscretePhoton H.264 encoder est très bien équipé pour les applications à temps critique telles que la vidéoconférence, ainsi que d'autres types d'applications.

La latence d'encodage faible signifie également une variation faible latence, ce qui pourrait entraîner extrêmement faible trame baisse de taux pour l'encodage direct.

Certains chiffres de performances peuvent être trouvés à partir de [the latest video codec comparison by MSU](#).

Pour plus d'informations et des versions d'évaluation, visitez www.discretephoton.com.

Utilisation de GraphEdt.exe



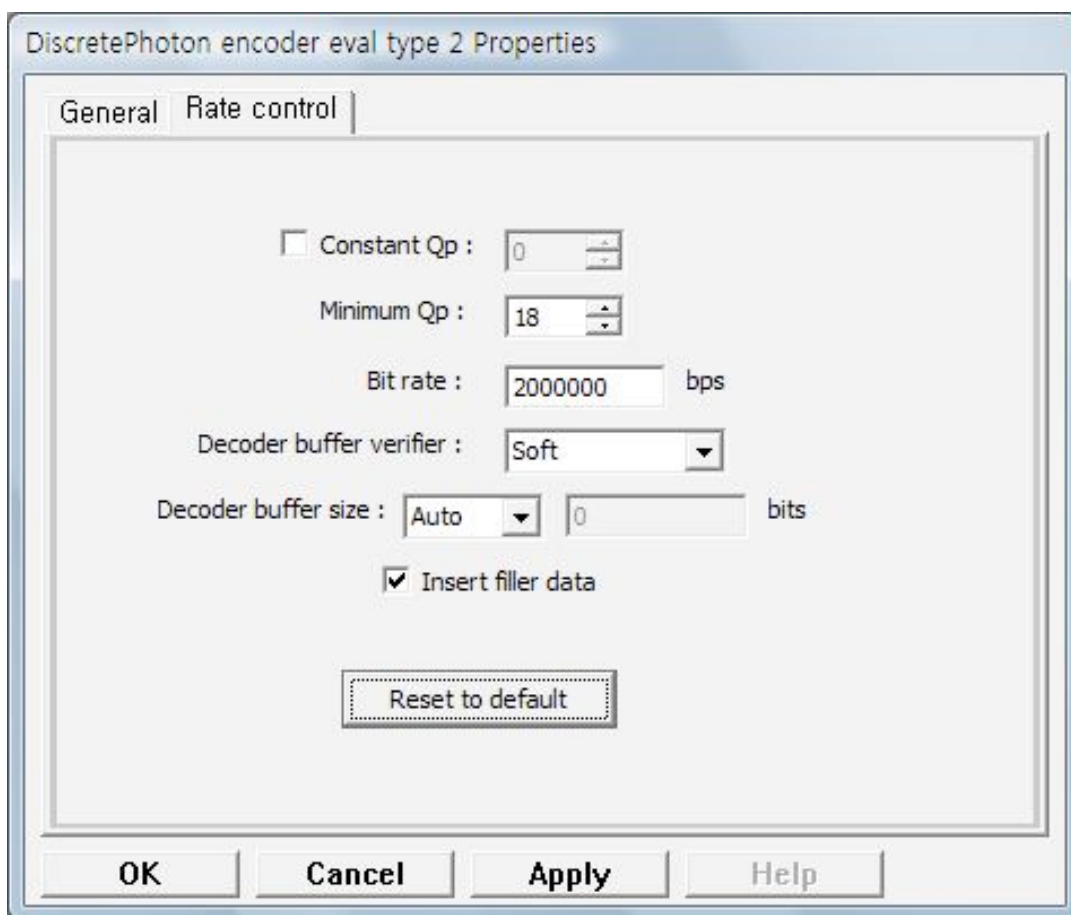
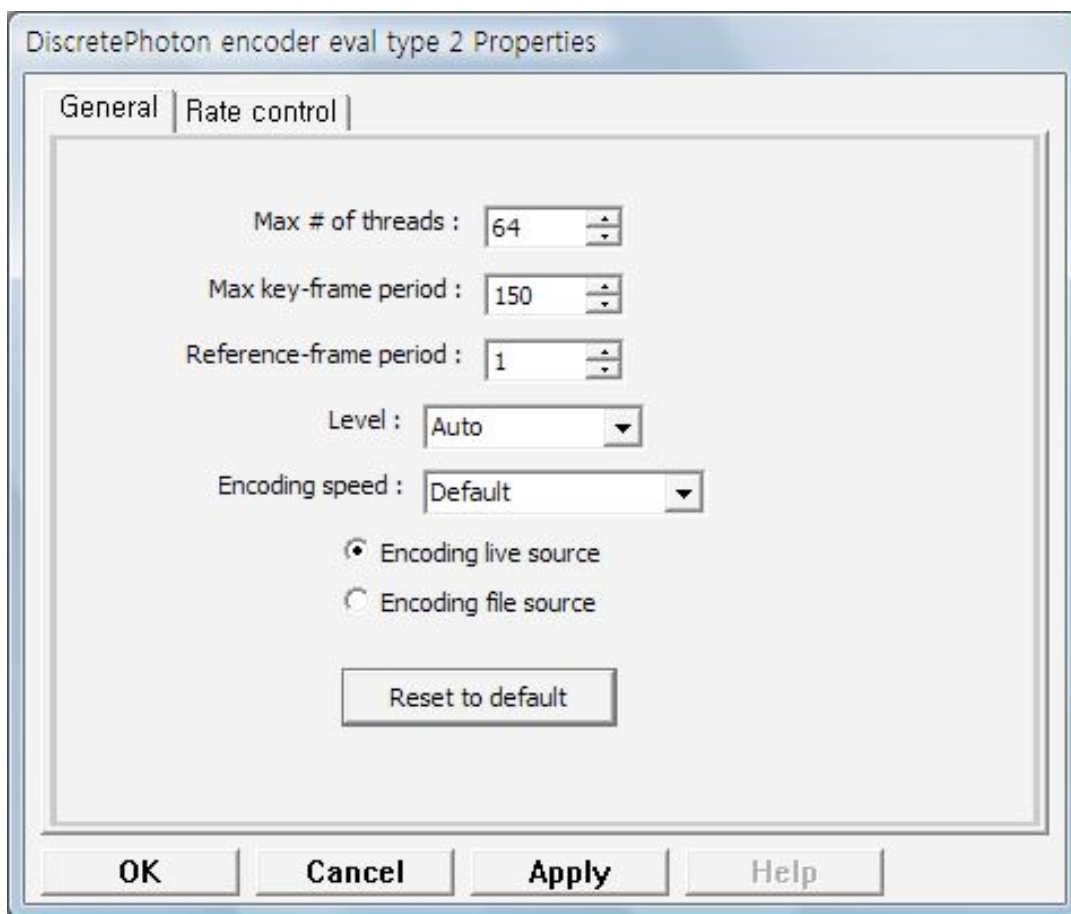
GraphEdt.exe est contenue dans Windows SDK. Vous pouvez l'utiliser pour construire et tester DirectShow filter graphique visuellement.

Après l'installation de la version 32-bit de l'encodeur H.264 DiscretePhoton, vous pouvez le trouver dans GraphEdt.exe à Graph -> Insert Filters... -> Video Compressors.

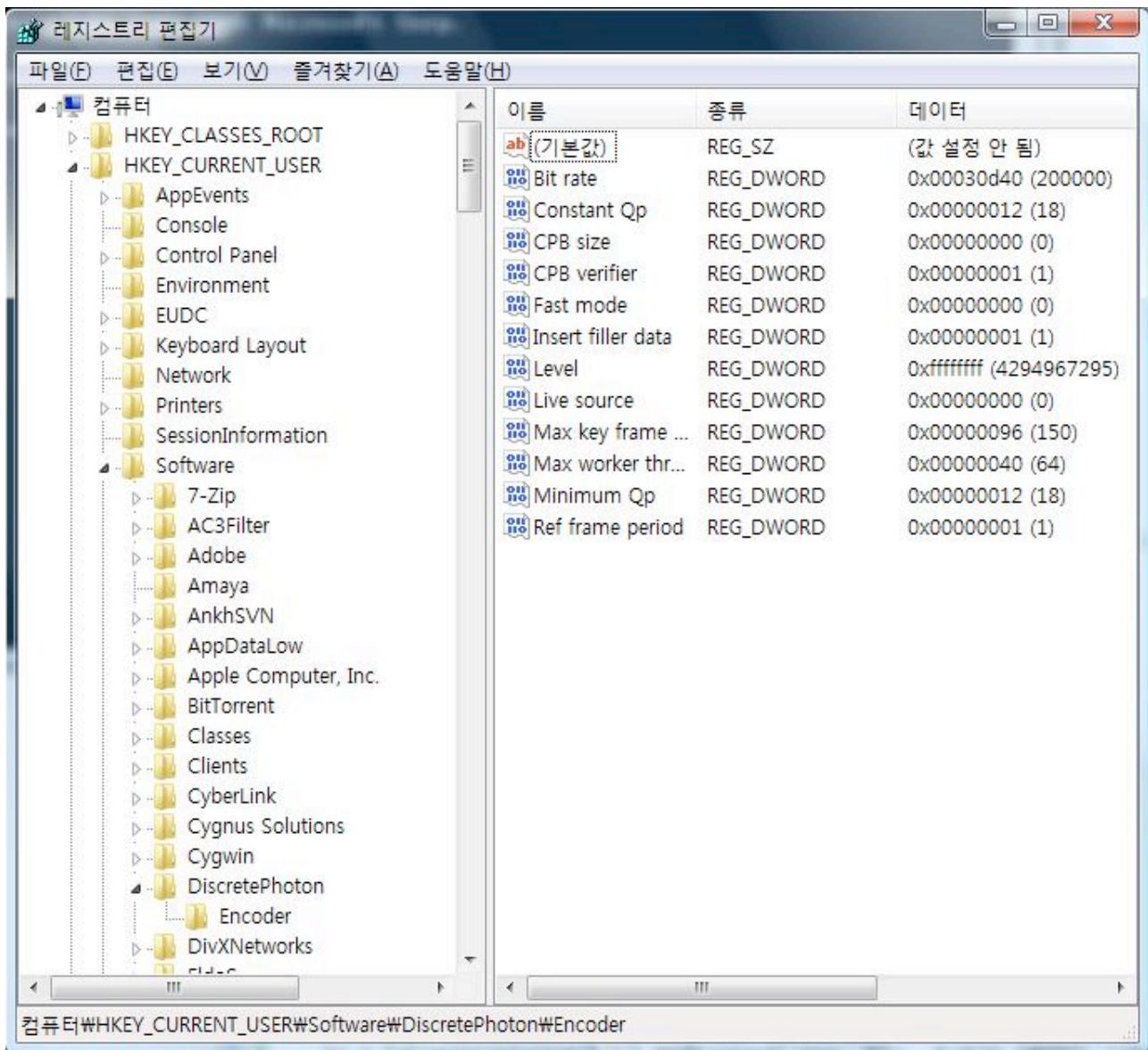
La couleur verte de DiscretePhoton H.264 encoder noeud indique qu'il n'est pas réellement un filtre DirectShow, mais une DMO (DirectX Media Object) avec enveloppement DirectShow.

DiscretePhoton H.264 encoder accepte **I420**, **YV12**, **UYVY**, **YUYV**, **RGB24**, **RGB32** comme format vidéo en entrée. Et il produit H.264 byte stream (FourCC:**H264**).

Si vous faites un clic droit sur le noeud DiscretePhoton H.264 encoder, page de propriétés seront présentées comme suit.



Il ya deux onglets: «General» et «Rate Control». Après avoir changé toute valeur et de l'appliquer, la valeur modifiée se souviendra dans le Registre Windows (à HKEY_CURRENT_USER\Software\DiscretePhoton\Encoder). Vous pouvez lire ou écrire ces valeurs de Registre dans votre programme.



Signification de chaque champ valeur est la suivante.

Max # of threads: ("Max worker threads" sur le registry.) Nombre maximum de threads qui peuvent être utilisés pendant le processus d'encodage. Le nombre n'est pas réel, mais au maximum parce que le nombre réel sera déterminé par le nombre de CPU-cores (ou hyper-thread) de votre système et la taille du cadre. Vous pouvez diminuer ce nombre en dessous de votre CPU-cores si vous ne voulez pas utiliser 100% de la puissance du CPU pendant le processus d'encodage.

Max key-frame period: ("Max key frame period" sur le registry.) Ici, key-frame signifie trame IDR de la norme H.264. Sa valeur est aussi maximale et

non réel, parce que les clés réelles-cadres pourraient apparaître plus fréquemment pour l'efficacité d'encodage. Minimum key-frame période est la moitié de cette valeur.

Reference-frame period: ("Ref frame period" sur le registry.) Par défaut, chaque image sera utilisée comme cadre de référence. Mais en augmentant cette valeur (en conséquence, non pas chaque frame sera utilisée comme reference frame) et en diminuant Max key-frame période, la tolérance de panne lors de la transmission de vidéo par réseau non fiable (comme UDP) pourrait être atteinte avec le coût de la diminution de compression ratio ou la qualité globale.

Level: (Sur le registry, -1:Auto, 0:Level_1, 1:Level_1b, ...) Il est encodage H.264 niveau (comme indiqué dans l'annexe A.3 de la spécification). Normalement, vous pouvez le définir comme «Auto». Puis le niveau sera déterminé automatiquement avec d'autres paramètres et d'entrée vidéo.

Encoding speed: ("Fast mode" sur le registry. 0:Default, 1:Fast, 2:Even faster) Vitesse encodage plus rapide peut être obtenue en sacrifiant une certaine qualité vidéo. Donc, si vous utilisez la plutôt à haut sortie bit-rate et que vous voulez accélérer un encodage plus rapide, vous pouvez le définir comme «Fast» ou «Even faster». Mais si vous utilisez la à faible sortie bit-rate, alors le «Even faster» mode peut produire en polyédrique vidéo.

Encoding live source / Encoding file source: ("Live source" sur le registry. 0:file source, 1:live source) Si votre vidéo source est en direct (comme celle d'une webcam ou une carte tuner TV, etc), choisissez 'Encoding live source'. Sinon, choisissez 'Encoding file source'.

Constant Qp: (Sur le registry, la valeur 0 le désactive.) Si vous définissez cette valeur, la qualité de sortie vidéo sera presque constant tout au long de sortie vidéo entière, mais la sortie à bit-rate fluctue sévèrement. C'est, de sortie à bit-rate binaire ne sera pas contrôlé du tout. Donc, il n'est pas bon pour la vidéo qui est transmis sur le réseau. La valeur valide varie de 18 - 51. Une valeur plus grande signifie plus de compression-ratio et la qualité vidéo inférieure.

Minimum Qp: Qp (quantization parameter) des valeurs qui sont inférieures à ce ne sera pas appliquée lors de rate-control.

Bit rate: Sortie bit-rate en bits par seconde.

Decoder buffer verifier: ("CPB verifier" sur le registry. 0:Hard, 1:Soft, 2:None) Il définit la rigidité du decoder buffer verifier. Ici, le decoder buffer correspond à la CPB (coded picture buffer) de l'annexe C de la norme H.264 spécification. (Avis: même «Hard» ne veut pas dire parfait).

Decoder buffer size: ("CPB size" sur le registry.) Taille en bits. Si elle est réglée sur «Auto», il sera fixé à la valeur maximale autorisée (de VCL HRD).

Insert filler data: Si decoder buffer est proche de overflow, et Qp atteint près de sa valeur minimale, filler data peuvent être insérés dans le flux de bits de sortie pour soutenir la sortie de débit binaire. Donc, normalement, il devrait être réglé.

Programmation avec DirectShow

Je crois qu'il ya quelques bonnes références sur la programmation en environnement DirectShow.

Vous pouvez également télécharger des exemples de programmes simples qui font usage de DiscretePhoton H.264 encoder de <http://www.discretephoton.com/php/downloader.php?f=Examples.zip> pour 32-bit version ou à partir d'http://www.discretephoton.com/php/downloader.php?f=Examples_x64.zip pour 64-bit version.

Pour le bâtiment C++ version, bibliothèque DirectShow classe de base est nécessaire, qui est contenu dans Windows SDK. Et C # version nécessite DirectShow.Net. Plus d'informations peuvent être trouvées dans README.htm.