

# Notes de mise à jour LabVIEW™

## Version 8.2

Ces notes de mise à jour décrivent le processus de mise à jour de LabVIEW pour Windows, Mac OS et Linux vers la version 8.2, les problèmes que vous pourriez rencontrer lors de cette mise à jour, ainsi que les nouvelles fonctionnalités.

Si vous mettez LabVIEW 7.1 ou une version antérieure au niveau de LabVIEW 8.2, reportez-vous aux *Notes de mise à jour LabVIEW 8.0* pour obtenir des informations sur les améliorations, les modifications et les nouvelles fonctionnalités de LabVIEW 8.0. National Instruments conseille aux utilisateurs qui mettent LabVIEW 7.1 ou une version antérieure à niveau de lire la section *Fonctionnalités et changements dans LabVIEW 8.0* des *Notes de mise à jour LabVIEW 8.0* en plus des présentes notes de mises à jour. Reportez-vous au site Web de National Instruments sur [ni.com/fr/info](http://ni.com/fr/info) et entrez l'info-code `rdfup8` pour accéder aux *Notes de mise à jour LabVIEW 8.0*.

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations supplémentaires sur les nouvelles fonctionnalités de LabVIEW 8.2, ainsi que des informations sur les concepts de programmation LabVIEW, des instructions pas à pas sur l'utilisation de LabVIEW et des informations de référence sur notamment les VIs, les fonctions, les palettes, les menus, les outils, les propriétés, les méthodes, les événements et les boîtes de dialogue de LabVIEW. L'*Aide LabVIEW* liste également les ressources de documentation disponibles chez National Instruments. Accédez à l'*Aide LabVIEW* en sélectionnant **Aide»Rechercher dans l'Aide LabVIEW**.

## Sommaire

---

Mise à jour vers LabVIEW 8.2 .....	2
Conversion des VIs .....	2
Mise à jour des toolkits, des drivers d'instruments et des logiciels complémentaires .....	3
Mise à jour des autres logiciels National Instruments .....	4
Mise à jour à partir de versions précédentes de LabVIEW .....	5

Problèmes de mise à jour et de compatibilité .....	6
Mise à jour à partir de LabVIEW 8.0 .....	6
Mise à jour à partir de LabVIEW 7.x .....	13
Mise à jour à partir de LabVIEW 6.x .....	34
Mise à jour à partir de LabVIEW 5.x ou versions antérieures .....	40
Fonctionnalités et changements dans LabVIEW 8.2 .....	40
Documentation de LabVIEW .....	41
Nouveaux VIs d'exemple .....	41
Améliorations apportées au temps de lancement .....	41
Améliorations apportées au diagramme .....	41
Améliorations apportées à la face-avant.....	43
Améliorations apportées à l'environnement.....	46
Améliorations apportées aux nœuds ; nouveau et modifications apportées aux VIs et fonctions.....	52
Nouveaux événements, propriétés et méthodes.....	62
Améliorations apportées à MathScript LabVIEW .....	62
Commande image 3D .....	65
Programmation orientée objet LabVIEW.....	66
Améliorations apportées aux projets LabVIEW.....	68
Contrôle à distance des VIs à partir de plusieurs clients .....	74
Importation de fonctions à partir d'un fichier de bibliothèque partagée .....	74
Modèles de drivers d'instruments .....	75
Améliorations apportées aux contrôles .NET et ActiveX (Windows).....	75
Améliorations apportées à l'Outil de recherche d'exemples NI .....	76
Améliorations apportées au contrôle de source.....	76
Améliorations apportées au format TDM.....	78
Importation de services Web (Windows).....	78
Modifications apportées aux fonctions pour le code externe .....	79

## Mise à jour vers LabVIEW 8.2

---

Si vous effectuez une mise à jour à partir d'une version précédente de LabVIEW, lisez d'abord cette section, *Mise à jour vers LabVIEW 8.2* et les sections *Mise à jour à partir de LabVIEW x.x* dans la section *Problèmes de mise à jour et de compatibilité* de ce document, où x.x est la version de LabVIEW que vous mettez à jour.

### Conversion des VIs

Lorsque vous ouvrez un VI enregistré dans LabVIEW 4.0 ou dans une version ultérieure, LabVIEW 8.2 convertit et compile automatiquement ce VI. Vous devez enregistrer le VI dans LabVIEW 8.2. Sinon, le processus de conversion, qui utilise des ressources mémoire supplémentaires de votre système, se produit à chaque fois que vous accédez à ce VI.

De plus, vous risquez d'avoir affaire à une dégradation des performances pour chaque VI contenant des modifications non enregistrées, pouvant même entraîner une recompilation. Reportez-vous à l'*Aide LabVIEW* pour en savoir plus sur les problèmes de performance et de mémoire.



**Remarque** Les VIs enregistrés dans LabVIEW 8.2 ne peuvent pas être chargés dans une version antérieure de LabVIEW. Sélectionnez **Fichier»Enregistrer pour la version précédente** pour enregistrer des VIs afin qu'ils puissent s'exécuter dans LabVIEW 8.0. Avant d'enregistrer les VIs dans LabVIEW 8.2, conservez une copie de sauvegarde des VIs que vous comptez utiliser dans LabVIEW 8.0 ou une version antérieure.

Si votre ordinateur ne possède pas suffisamment de mémoire pour convertir tous les VIs simultanément, convertissez-les par étapes. Examinez la hiérarchie des VIs que vous voulez convertir et commencez par charger et enregistrer les sous-VIs des niveaux inférieurs de la hiérarchie. Vous pouvez ensuite remonter progressivement aux niveaux supérieurs. Le VI de niveau principal doit être ouvert et converti en dernier. Vous pouvez aussi sélectionner **Outils»Avancé»Recompiler** pour convertir un répertoire de VIs. Cependant, la recompilation convertit les VIs situés dans un répertoire ou une LLB par ordre alphabétique. Si le processus de conversion rencontre d'abord un VI de niveau principal, la recompilation nécessite approximativement autant de mémoire que si vous ouvriez d'abord le VI de niveau principal.

Vous pouvez surveiller l'utilisation de la mémoire en sélectionnant **Aide» À propos de LabVIEW** pour afficher un résumé de la quantité de mémoire en cours d'utilisation.

## Mise à jour des toolkits, des drivers d'instruments et des logiciels complémentaires

Après avoir installé LabVIEW 8.2, assurez-vous que vos toolkits et logiciels supplémentaires sont compatibles avec LabVIEW 8.2 puis réinstallez-les dans le répertoire LabVIEW 8.2. Il faudra peut-être que vous désinstalliez le toolkit prévu pour une version précédente de LabVIEW. Reportez-vous à la documentation qui accompagne votre toolkit ou logiciel supplémentaire pour obtenir la procédure d'installation exacte.



**Remarque** Les versions des modules de LabVIEW doivent correspondre à la version de LabVIEW. Par exemple, vous devez utiliser le module LabVIEW Real-Time 8.2 avec LabVIEW 8.2.

Vous devez en outre compiler les VIs des toolkits, drivers d'instruments et logiciels supplémentaires existants pour pouvoir les utiliser dans LabVIEW 8.2. Reportez-vous à la section *Conversion des VIs* de ce

document pour obtenir des informations plus détaillées sur la recompilation des VIs.

Les toolkits, les drivers d'instruments et les logiciels supplémentaires suivants nécessitent des mises à jour ou des téléchargements pour être utilisés dans LabVIEW 8.2 :

- Si vous avez l'Application Builder de LabVIEW, vous devez le mettre à jour avec la version 8.2. Le système de développement professionnel LabVIEW 8.2 comprend l'Application Builder 8.2. Reportez-vous au fichier *Fichier Readme de l'Application Builder de LabVIEW* situé dans le répertoire `labview\readme` de **(Windows)** ou sous le répertoire `labview` de **(Mac OS et Linux)** pour obtenir davantage d'informations sur l'installation de l'Application Builder de LabVIEW.
- Vous devez utiliser le VI Analyzer 1.1 dans LabVIEW 8. x. Reportez-vous au site Web de National Instruments sur [ni.com/frinfo](http://ni.com/frinfo) et entrez l'info-code `exd8yy` pour accéder au Guide interactif de mise à jour et acheter VI Analyzer 1.1.
- Vous devez télécharger des VIs supplémentaires pour utiliser Internet Toolkit 6.0 avec LabVIEW 8.x. Reportez-vous au site Web de National Instruments sur [ni.com/frinfo](http://ni.com/frinfo) et entrez l'info-code `itkver6` pour télécharger les VIs nécessaires.
- Le driver d'instruments du multimètre numérique HP/Agilent 34401A (DMM) est maintenant plus proche du driver DMM modèle de National Instruments. Ce driver n'est pas compatible avec le driver HP34401A qui était fourni avec LabVIEW 7.x et les versions antérieures. Pour avoir un driver compatible avec le driver HP34401A LabVIEW 7.x, téléchargez ce driver-là de la base de drivers d'instruments NI, à l'adresse [ni.com/idnet](http://ni.com/idnet).

## Mise à jour des autres logiciels National Instruments

Vous devez utiliser NI TestStand 3.5 ou une version ultérieure avec LabVIEW 8.x. Reportez-vous au site Web de National Instruments à l'adresse [ni.com/frinfo](http://ni.com/frinfo) et entrez l'info-code `exd8yy` pour accéder au Guide interactif de mise à jour et acheter NI TestStand 3.5 ou une version ultérieure.

## Mise à jour à partir de versions précédentes de LabVIEW

Les mises à jour vers de nouvelles versions de LabVIEW n'ont aucun effet sur les versions antérieures de LabVIEW installées sur l'ordinateur car les nouvelles versions s'installent dans un autre répertoire. LabVIEW 5.x et les versions antérieures s'installent dans le répertoire `labview`. LabVIEW 6.0 et les versions ultérieures s'installent dans le répertoire `labviewx.x`, où `x.x` correspond au numéro de version. Vous pouvez installer LabVIEW 8.2 sans désinstaller les versions précédentes de LabVIEW.

## Remplacement d'une version existante de LabVIEW

Pour remplacer votre version existante de LabVIEW, désinstallez-la, exécutez l'installeur de LabVIEW 8.2 et définissez le répertoire d'installation au même répertoire `labview` que celui dans lequel vous avez installé la version précédente de LabVIEW.

**(Windows)** Vous pouvez aussi remplacer la version existante de LabVIEW par LabVIEW 8.2 au moyen de l'applet Ajout/Suppression de programmes du Panneau de configuration afin de désinstaller la version existante de LabVIEW. Le programme de désinstallation ne supprime pas les fichiers que vous avez créés dans le répertoire `labview`.



**Remarque** Lorsque vous désinstallez ou réinstallez LabVIEW, LabVIEW désinstalle les fichiers de format `.llb` du répertoire `vi.lib`, y compris tout VI ou commande que vous avez enregistrés dans les fichiers `.llb`. Enregistrez vos VIs et commandes dans le répertoire `user.lib` pour les ajouter aux palettes **Commandes** et **Fonctions**.

## Copie des paramètres d'environnement d'une version précédente de LabVIEW

Pour utiliser les paramètres de l'environnement LabVIEW d'une version précédente de LabVIEW, copiez le fichier de préférences de LabVIEW du répertoire `labview` dans lequel la version précédente est installée.



**Attention** Si vous remplacez le fichier de préférences de LabVIEW 8.2 par un fichier de préférences d'une version antérieure, vous risquez d'écraser les définitions de préférences qui ont été ajoutées à LabVIEW depuis la version précédente.

Après avoir installé LabVIEW 8.2, copiez le fichier de préférences LabVIEW dans le répertoire LabVIEW 8.2.

**(Windows)** LabVIEW enregistre les préférences dans le fichier `labview.ini`.

**(Mac OS)** LabVIEW enregistre les préférences dans le fichier `LabVIEW Preferences` du dossier `Library:Preferences` situé dans votre répertoire “home”.

**(Linux)** LabVIEW enregistre les préférences dans le fichier `.labviewrc` de votre répertoire “home”.

## Copie de fichiers `user.lib` à partir d'une version antérieure de LabVIEW

Pour utiliser les fichiers du répertoire `user.lib` d'une version précédente de LabVIEW, copiez les fichiers à partir du répertoire `labview` dans lequel la version précédente est installée. Après avoir installé LabVIEW 8.2, copiez les fichiers dans le répertoire `user.lib` du répertoire LabVIEW 8.2.

## Problèmes de mise à jour et de compatibilité

---

Reportez-vous aux sections qui suivent pour connaître les problèmes de mise à niveau et de compatibilité spécifiques aux différentes versions de LabVIEW.

Reportez-vous au fichier `readme.html` du répertoire `labview` pour obtenir des informations plus détaillées sur les problèmes connus, les problèmes de compatibilité et pour obtenir des informations sur des fonctionnalités ajoutées tardivement dans LabVIEW 8.2.

### Mise à jour à partir de LabVIEW 8.0

Il est possible que vous rencontriez les problèmes de compatibilité suivants lorsque vous passez de LabVIEW 8.0 à LabVIEW 8.2.

### Plates-formes supportées

LabVIEW 8.2 comprend les changements suivants affectant les plates-formes supportées :

- LabVIEW 8.2 ne supporte pas Windows XP x64.
- LabVIEW 8.2 ne supporte pas Mac OS X 10.3.8 ou versions antérieures.
- LabVIEW 8.2 offre un support pour les ordinateurs Macintosh dotés de processeurs Intel. Reportez-vous au site Web de National Instruments sur [ni.com/frinfo](http://ni.com/frinfo) et entrez l'info-code `macintel` pour obtenir davantage d'informations sur le support pour Macintosh.

## Configuration système requise

**(Mac OS)** LabVIEW 8.2 exige au minimum 500 Mo d'espace disque pour l'installation minimale de LabVIEW ou 700 Mo pour l'installation complète.

**(Linux)** LabVIEW 8.2 exige au minimum 430 Mo d'espace disque pour l'installation minimale de LabVIEW ou 620 Mo pour l'installation complète.

## Documentation imprimée

Les documents suivants n'ont pas été modifiés pour LabVIEW 8.2. Leur contenu ne reflète donc pas forcément les modifications apportées à LabVIEW 8.2.

- *Carte de référence rapide de LabVIEW*
- *Principes de base de LabVIEW* — Dans la mesure où les *Principes de base de LabVIEW* constituent une sous-rubrique du livre **Principes de base** de l'*Aide LabVIEW*, reportez-vous au livre **Principes de base** de l'onglet **Sommaire** dans l'*Aide LabVIEW* pour consulter le contenu mis à jour.

## Modification du comportement de plusieurs VIs et fonctions

Le comportement des fonctions et VIs suivants a changé dans LabVIEW 8.2.

### Communication entre plusieurs instances d'application

Dans LabVIEW 8.2, vous ne pouvez pas utiliser les fonctions Obtenir une file d'attente, Obtenir un notificateur, Créer un événement utilisateur, Créer un sémaphore et Créer un rendez-vous pour communiquer entre différentes instances d'application. Si vous obtenez ou créez une référence de file d'attente, de notificateur, d'événement utilisateur, de sémaphore ou de rendez-vous dans une instance d'application, vous ne pouvez pas utiliser cette référence dans une autre instance d'application.

### VI Transformation inverse des vecteurs propres

Les entrées **action**, **indice bas**, **indice haut** et **Échelle** du VI Transformation inverse des vecteurs propres sont devenues des entrées requises.

## Fonction DataSocket Écrire

Dans LabVIEW 8.0.1, le comportement par défaut de la fonction DataSocket Écrire est asynchrone. Si LabVIEW 8.0 et LabVIEW 8.2 sont installés sur votre ordinateur, le VI d'exemple DataSocket API Client du répertoire `labview\examples\Shared Variable` renvoie une erreur quand vous arrêtez le VI. Vous devez mettre à jour LabVIEW 8.0 à la version LabVIEW 8.0.1 pour pouvoir utiliser cet exemple dans Labview 8.2.

## VI d'E/S sur fichiers

Les VI Écrire dans un fichier tableur et Lire un fichier tableur sont des VI polymorphes. Le VI Écrire dans un fichier tableur s'adapte à la valeur câblée à l'entrée **format**. Le VI Lire un fichier tableur comprend les instances suivantes : DBL, I64 et chaîne.

## Fonction GPIB Status

Dans LabVIEW 8.0, la fonction GPIB Status ne s'exécutait pas si l'entrée **entrée d'erreur** recevait une erreur. Dans LabVIEW 8.2, la fonction GPIB Status s'exécute toujours, même si l'entrée **entrée d'erreur** reçoit une erreur.

## VI Histogramme

La valeur par défaut de l'entrée **intervalles** du VI Histogramme est devenue 10.

## Fonction Ouvrir une référence de VI

Le comportement par défaut de l'entrée **options** de la fonction Ouvrir une référence de VI est de demander aux utilisateurs de rechercher les sous-VI manquants du VI référencé. Une nouvelle valeur, `0x20`, spécifie qu'il ne faut ni afficher la boîte de dialogue **Rechercher** ni demander aux utilisateurs de rechercher les sous-VI manquants du VI référencé.

## VI Racines d'un polynôme

Si  $P(x)$  est égal à une constante différente de zéro, le VI Racines d'un polynôme ne renvoie pas d'erreur. Toutefois, si  $P(x)$  est égal à 0, le VI Racines d'un polynôme renvoie l'erreur -20111. Les coefficient polynomiaux en entrée de ce VI ne peuvent pas tous être 0.

## VI Motif de rampe

Dans le VI Motif de rampe, si **échantillons** est égal à 1 et que **exclure la fin ?** est VRAI, le VI renvoie un tableau contenant un élément de **début**, sans erreurs. Dans LabVIEW 8.0, le VI renvoie une erreur dans de telles conditions.

## VI Lire une valeur de registre type simple

LabVIEW 8.0 ne gérait par correctement le formatage de chaîne `REG_MULTI_SZ` que le VI utilisait pour un tableau aplati de chaînes. Il vous fallait définir une analyse syntaxique pour gérer ce type de données pour le VI Lire une valeur de registre type simple. Dans LabVIEW 8.2, le VI Lire une valeur de registre type simple renvoie les données dans le format utilisé dans le VI Écrire une valeur de registre type simple. Vous n'êtes plus obligé de définir l'analyse syntaxique. Si vous utilisez votre propre analyseur avec ces VIs dans LabVIEW 8.2, le VI Lire une valeur de registre type simple renvoie de mauvaises données.

## VI Rééchantillonner les waveforms (opération unique)

La valeur par défaut de l'entrée **intervalle ouvert ?** du VI Rééchantillonner une waveform (opération unique) est passée de VRAI à FAUX, ce qui sélectionne un intervalle fermé. Si vous ne mettez pas à jour du code existant en conséquence, il se peut que le VI ne renvoie pas le résultat attendu.

## VIs de son

Dans les VIs Lire une entrée de son et Lire un fichier son - simple, l'élément **t0** de la sortie **données** renvoie l'horodatage du premier échantillon lu. LabVIEW évalue l'heure initiale à laquelle il a lu le premier échantillon.

Il n'est plus nécessaire d'appeler le VI Arrêter la sortie de son pour arrêter le son sur une tâche de son continue.

Arrêter une sortie de son fonctionne en mode **Échantillons continus** et en mode **Échantillons finis**.

## VIs waveform

Les modifications suivantes ont été apportées aux VIs waveform dans LabVIEW 8.2 :

- VI Détection élémentaire du niveau de déclenchement — Dans les deux instances de ce VI, l'entrée **pente** s'appelle désormais **pente de déclenchement**.

- VI Sous-ensemble d'une waveform — Comporte les instances suivantes : WDT Sous-ensemble d'une waveform DBL, WDT Sous-ensemble d'une waveform CDB, WDT Sous-ensemble d'une waveform EXT, WDT Sous-ensemble d'une waveform I16, WDT Sous-ensemble d'une waveform I32, WDT Sous-ensemble d'une waveform I8 et WDT Sous-ensemble d'une waveform SGL. L'entrée **format de début/durée** n'a plus l'option **Temps absolu**. L'entrée **début** a été renommée **temps/échantillons de départ** et la sortie **début réel** a été renommée **temps/échantillons de départ réels**.
- VI Tableau des temps d'une waveform — Le type de données de la sortie **tableau X** est passé de nombre à virgule flottante double-précision à horodatage.
- VI Valeur ponctuelle de Y — Ce VI et les instances polymorphes correspondantes ont été renommés Obtenir une valeur XY. Le VI Obtenir une valeur XY comporte désormais une sortie **valeur de X** et la sortie **valeur des données** s'appelle désormais **valeur de Y**.
- VI Nombre d'échantillons d'une waveform — Ce VI est un VI polymorphe doté des instances suivantes : WDT Nombre d'échantillons d'une waveform DBL, WDT Nombre d'échantillons d'une waveform CDB, WDT Nombre d'échantillons d'une waveform EXT, WDT Nombre d'échantillons d'une waveform I16, WDT Nombre d'échantillons d'une waveform I32, WDT Nombre d'échantillons d'une waveform I8 et WDT Nombre d'échantillons d'une waveform SGL.
- VI Lire une waveform à partir d'un fichier — Renvoie un état d'erreur VRAI dans la sortie **sortie d'erreur** lorsque l'erreur est fin de fichier.
- VI Remplacer un sous-ensemble — L'entrée **début** a été renommée **temps/échantillons de départ** et la sortie **valeur de début réelle** a été renommée **temps/échantillons de départ réels**.
- VI Rechercher un pattern numérique — L'entrée **début** a été renommée **temps/indice de départ**.
- VI Chercher une waveform — Le type de données des sorties **temps du meilleur ajustement** et **temps des ajustements** est passé de nombre à virgule flottante double-précision à horodatage.
- VI Min et max d'une waveform — Le type de données des sorties **temps min** et **temps max** est passé de nombre à virgule flottante double-précision à horodatage.
- VI Waveform en paires XY — Le type de données de l'élément **x** de la sortie **paires XY** est passé de nombre à virgule flottante double-précision à horodatage.

# Modification du comportement de propriétés, méthodes et événements

Le comportement des propriétés, méthodes et événements suivants a changé dans LabVIEW 8.2 :

- Le comportement par défaut de l'entrée **options** de la méthode ActiveX GetVIREference est de demander aux utilisateurs de rechercher les sous-VIs manquants du VI référencé. Une nouvelle valeur, 0x20, spécifie qu'il ne faut ni afficher la boîte de dialogue **Rechercher** ni demander aux utilisateurs de rechercher les sous-VIs manquants du VI référencé.
- La méthode Ajouter un élément de la classe ÉlémentProjet (ProjectItem pour ActiveX) renvoient une erreur si vous essayez d'ajouter une variable partagée à une bibliothèque qui n'est pas ouverte dans un projet.
- Si vous définissez le paramètre **Supprimer réf. autom.** de la méthode Exécuter le VI à VRAI et que la méthode renvoie une erreur, LabVIEW supprime la référence.
- Les valeurs valides pour la propriété Application : langue comporte le paramètre zh-cn pour indiquer que l'environnement LabVIEW est en chinois simplifié.
- Dans LabVIEW 8.0, les méthodes .NET qui passent des types de données tableau par référence passent toutes les données avec le type de données refnum. Les méthodes .NET qui passent des types de données tableau par référence passent les données avec le type de données véritable.
- La propriété Éditer la position des classes TableNumériques, ListeMultiColonne, Table et CommandeArbre renvoie les valeurs (-2, -2) pour indiquer qu'aucune modification n'est apportée au texte de la commande. La propriété Éditer la ligne de la classe ListeDéroulante renvoie la valeur -2 pour indiquer qu'aucune modification n'est apportée au texte de la commande.
- Dans LabVIEW 8.0, la propriété Retarder MàJ face-avant ne faisait pas référence à la mise à jour des faces-avant dans une face-avant secondaire. Dans LabVIEW 8.2, la propriété Retarder MàJ face-avant fonctionne avec les faces-avant secondaires.
- Les événements Fermeture d'instance d'app. et Fermeture d'instance d'app. ? remplacent les événements Sortie de l'application ? et Sortie de l'application. Lorsque vous utilisez l'événement Fermeture d'instance d'app. dans un VI s'exécutant hors d'un projet LabVIEW, LabVIEW génère l'événement quand vous quittez LabVIEW par l'interface utilisateur ou par programmation. LabVIEW génère l'événement Fermeture d'instance d'app. ? lorsque vous quittez LabVIEW par l'interface utilisateur. Lorsque vous vous enregistrez les

événements Fermeture d'instance d'app. ? et Fermeture d'instance d'app. pour un VI qui s'exécute dans un projet LabVIEW, LabVIEW génère les événements quand l'instance d'application se ferme ou quand vous quittez LabVIEW.

## Propriétés, méthodes et événements obsolètes

LabVIEW 8.2 ne supporte pas les propriétés, méthodes et événements suivants.

- LabVIEW 8.2 ne supporte pas la propriété Connecteur.
- LabVIEW 8.x ne supporte pas la propriété Type de données pour la classe Variable. À la place, utilisez la propriété Type de données (variant) de la classe Variable.

## Propriétés, méthodes et événements renommés

Les propriétés, méthodes et événements suivants ont été renommés dans LabVIEW 8.2 :

Classe	Nom dans LabVIEW 8.0	Nom dans LabVIEW 8.2	Type
Application	Déconnecter de l'esclave	LVRT : déconnecter de l'esclave	Méthode
Application	Sortie de l'application	Fermeture d'instance d'application	Événement
Application	Sortie de l'application ?	Fermeture d'instance d'app. ?	Événement
GrapheIntensité, GrapheSignauxMixte et Graphe	Palette de curseur : visible	Légende des curseurs visible	Propriété
Bibliothèque	Supprimer un tag de bibliothèque	Supprimer le tag de bibliothèque	Méthode
Bibliothèque	Obtenir l'icône	Icône : obtenir	Méthode
Bibliothèque	Obtenir un tag de bibliothèque	Tag de bibliothèque : obtenir	Méthode
Bibliothèque	Obtenir les noms des tags de bibliothèque	Tags de bibliothèque : obtenir les noms	Méthode
Bibliothèque	Obtenir l'état de verrouillage	État de verrouillage : obtenir	Méthode

Classe	Nom dans LabVIEW 8.0	Nom dans LabVIEW 8.2	Type
Bibliothèque	Obtenir l'étendue de la source	Étendue de la source : obtenir	Méthode
Bibliothèque	Enregistrer	Enregistrer : bibliothèque	Méthode
Bibliothèque	Enregistrer une copie	Enregistrer : copie	Méthode
Bibliothèque	Définir l'icône	Icône : définir	Méthode
Bibliothèque	Définir un tag de bibliothèque	Tag de bibliothèque : définir	Méthode
Bibliothèque	Définir l'état de verrouillage	État de verrouillage : définir	Méthode
Bibliothèque	Définir l'étendue de la source	Étendue de la source : définir	Méthode
ListeDéroulante, ListeMultiColonne et CommandeArbre	Glisser/Déposer : autoriser le glissement d'éléments	Glisser/Déposer : autoriser le glissement	Propriété
Chemin et Chaîne	Autoriser le dépôt	Autoriser le dépôt	Propriété
ÉlémentProjet	Supprimer un tag	Tag : supprimer	Propriété
ÉlémentProjet	Obtenir un tag	Tag : obtenir un tag	Propriété
ÉlémentProjet	Obtenir les noms des tags	Tag : obtenir les noms	Propriété
ÉlémentProjet	Obtenir un tag XML	Tag : obtenir un tag XML	Propriété
ÉlémentProjet	Définir un tag	Tag : définir un tag	Propriété
ÉlémentProjet	Définir un tag XML	Tag : définir un tag XML	Propriété
ÉlémentProjet	Chaîne du type de l'élément de bibliothèque	Type de l'élément de bibliothèque : chaîne	Propriété
ÉlémentProjet	Type de l'élément de bibliothèque	Élément de bibliothèque : type	Propriété

## Mise à jour à partir de LabVIEW 7.x

Il est possible que vous rencontriez les problèmes de compatibilité suivants lorsque vous passez de LabVIEW 7.x à LabVIEW 8.2. Reportez-vous à la section [Mise à jour à partir de LabVIEW 8.0](#) de ce document pour obtenir des informations sur d'autres problèmes de mise à jour que vous pourriez éventuellement rencontrer.

Reportez-vous aux *Notes de mise à jour LabVIEW* spécifiques à chaque version de LabVIEW entre les versions 7.x et 8.0, sur [ni.com/manuals](http://ni.com/manuals), pour obtenir des informations complémentaires sur les nouvelles fonctionnalités et modifications apportées à ces versions.



**Remarque** La *Carte de référence rapide LabVIEW* et le *Manuel des Principes de base de LabVIEW* n'ont pas été modifiés pour LabVIEW 8.2. Vous pouvez accéder aux versions PDF de ces documents dans le répertoire `labview\manuals`. Reportez-vous à la section [Mise à jour à partir de LabVIEW 8.0](#) de ce document pour obtenir de plus amples informations sur ces documents.

## Plates-formes supportées

Les modifications suivantes ont été apportées à LabVIEW 8.x en matière de plates-formes supportées :

- LabVIEW 7.1 et versions ultérieures ne sont pas prises en charge par Windows Me/98/95. LabVIEW 8.x n'est pas pris en charge par Windows NT.
- LabVIEW 8.x ne supporte pas Mac OS X 10.2 ou versions antérieures.
- LabVIEW 8.x n'est pas pris en charge par Sun Solaris.

## Configuration système requise

LabVIEW 7.x exige un minimum de 128 Mo de RAM, mais National Instruments recommande 256 Mo de RAM. LabVIEW 8.x exige un minimum de 256 Mo de RAM, mais National Instruments recommande 512 Mo de RAM.

LabVIEW 7.x exige une résolution d'écran de 800 × 600 pixels, mais National Instruments recommande une résolution d'écran de 1024 × 768 pixels. LabVIEW 8.x exige une résolution d'écran de 1024 × 768 pixels.

## Windows

LabVIEW 7.x exige au minimum un processeur Pentium III ou supérieur, ou un processeur Celeron 600 MHz ou équivalent, mais National Instruments recommande un processeur Pentium 4 ou équivalent. LabVIEW 8.x exige au minimum un processeur Pentium III ou Celeron 866 MHz ou équivalent, mais National Instruments recommande un processeur Pentium 4/M ou équivalent.

LabVIEW 7.x exige au moins 130 Mo d'espace disque pour l'installation minimale de LabVIEW ou 550 Mo pour l'installation complète. LabVIEW 8.x nécessite 1,2 Go d'espace disque pour l'installation complète de LabVIEW.

## Mac OS

LabVIEW 7.x exige au moins 280 Mo d'espace disque pour l'installation minimale de LabVIEW ou 350 Mo pour l'installation complète.

LabVIEW 8.2 exige au minimum 500 Mo d'espace disque pour l'installation minimale de LabVIEW ou 700 Mo pour l'installation complète.

## Linux

LabVIEW 7.x exige au minimum un processeur Pentium III ou supérieur, ou un processeur Celeron 600 MHz ou équivalent, mais National Instruments recommande un processeur Pentium 4 ou équivalent.

LabVIEW 8.x exige au minimum un processeur Pentium III ou Celeron 866 MHz ou équivalent, mais National Instruments recommande un processeur Pentium 4/M ou équivalent.

LabVIEW 7.x exige au moins 200 Mo d'espace disque pour l'installation minimale de LabVIEW ou 300 Mo pour l'installation complète.

LabVIEW 8.2 exige au minimum 430 Mo d'espace disque pour l'installation minimale de LabVIEW ou 620 Mo pour l'installation complète.

LabVIEW 7.x exige la version 2.1.3 ou une version ultérieure de la bibliothèque C GNU (`glibc`), mais National Instruments recommande la version 2.2.4 ou ultérieure. LabVIEW 8.x exige la bibliothèque C GNU version 2.2.4 ou une version ultérieure.

LabVIEW 7.x s'exécute sous Red Hat Linux 7.0 ou version ultérieure, Mandrake Linux 8.0 ou version ultérieure, SuSE Linux 7.1 ou version ultérieure, et Debian Linux 3.0 ou version ultérieure. LabVIEW 8.x s'exécute sous Red Hat Enterprise Linux WS 3 ou version ultérieure MandrakeLinux/Mandriva 10.0 ou version ultérieure, or SuSE Linux 9.1 ou version ultérieure.

## Types de palette personnalisés

LabVIEW 8.x ne supporte pas les types de palettes personnalisés. Vous pouvez éditer un jeu de palettes sans utiliser de type personnalisé.

Reportez-vous au site Web de National Instruments sur [ni.com/frinfo](http://ni.com/frinfo) et entrez l'info-code `lv8palette` pour obtenir davantage d'informations sur les modifications apportées aux palettes dans LabVIEW 8.0.

# Modification du comportement de plusieurs VIs et fonctions

Le comportement des fonctions et VIs suivants a changé dans LabVIEW 7.1 ou 8.0.

## Applications et VIs .NET

Il faut .NET Framework 1.1 Service Pack 1 ou une version ultérieure pour pouvoir utiliser les fonctions et applications .NET. dans LabVIEW 8.x. Vous devez supprimer le Hotfix KB886904 pour Microsoft .NET Framework 1.1 avant d'installer le .NET Framework 1.1 Service Pack 1.

Si vous chargez un VI .NET enregistré pour la dernière fois dans LabVIEW 7.x, il est possible que LabVIEW 8.x vous demande de rechercher les assemblies auxquels ce VI fait référence, même si les fichiers d'assembly se trouvent dans le même répertoire que le VI ou si vous les avez enregistrés en sélectionnant **Outils»Avancé»Références aux assemblies .NET** dans LabVIEW 7.x.

## Algorithmes des VIs d'analyse

Dans LabVIEW 7.1 et versions ultérieures, les VIs d'analyse utilisent les algorithmes BLAS/LAPACK. Désormais, ces VIs génèrent des résultats plus précis. Dans LabVIEW 8.x, ces VIs sont situés sur les palettes **Mathématiques** et **Traitement du signal**.

## VI Express Ajouter des signaux

Dans LabVIEW 7.x, si **Signal A en entrée** du VI Express Ajouter des signaux est vide ou non câblé et que vous câblez un seul signal ou un signal combiné à **Signal B en entrée**, la sortie de **Ajouter des signaux** est vide. Dans LabVIEW 8.x, si **Signal A en entrée** est vide ou n'est pas câblé, et que vous câblez un seul signal à **Signal B en entrée**, le VI Express renvoie **Signal B en entrée**. Si vous ne câblez qu'un signal combiné à **Signal B en entrée**, chaque signal du signal combiné ajoute le signal suivant à la fin pour créer un seul signal.

## Fonctions de comparaison

Dans LabVIEW 7.x et les versions antérieures, lorsque vous utilisez les fonctions de comparaison pour comparer des données variant, LabVIEW compare d'abord la longueur des deux variants puis compare les bits des variants. LabVIEW 8.x commence par comparer les codes de type des données variant, qui encodent les informations de type des variants, puis compare les autres attributs pour chaque type.

## VI Produit scalaire

Dans LabVIEW 7.0, le VI Produit scalaire calcule le produit scalaire des vecteurs d'entrée  $X$  et  $Y$  en utilisant l'équation suivante :

$$X*Y = \sum_{i=0}^{n-1} x_i y_i$$

Dans LabVIEW 7.1 et versions ultérieures, le VI Produit scalaire calcule le produit scalaire d'entrées complexes en utilisant l'équation suivante :

$$X*Y = \sum_{i=0}^{n-1} x_i y_i^*$$

où  $y_i^*$  est la conjuguée complexe de  $y_i$ .

## VI Easy Text Report (Mac OS et Linux)

Le connecteur du VI Easy Text Report a changé. Dans LabVIEW 8.x, lorsque vous ouvrez un VI qui a été enregistré pour la dernière fois dans LabVIEW 7.x ou une version antérieure qui utilise le VI Easy Text Report, vous devez cliquer avec le bouton droit sur le sous-VI et sélectionner **Relink To SubVI** dans le menu local.

## Fonction Formater en chaîne

Dans LabVIEW 7.x, l'utilisation du format %o ou %x en conjonction avec la fonction Formater en chaîne arrondit une entrée flottante en un entier 32 bits avant de la convertir en chaîne.

Dans LabVIEW 8.x, avec ces spécificateurs de format, la fonction arrondit les entrées flottantes en entiers 64 bits avant de les convertir en chaînes.

## Fonction Regrouper des nombres

Dans LabVIEW 7.x et versions antérieures, la fonction Regrouper des nombres contraint des entrées de type entier 32 bits à prendre la forme d'entiers 16 bits pour créer un entier 32 bits. Dans LabVIEW 8.x, la fonction Regrouper des nombres regroupe des entrées de type entier 32 bits pour former un entier 64 bits.



**Remarque** Si vous ouvrez un VI créé dans LabVIEW 7.x avec LabVIEW 8.x, LabVIEW contraint les entrées de type entier 32 bits à prendre la forme d'entiers 16 bits.

## Matrices et VIs mathématiques

Dans LabVIEW 8.x, les VIs **Mathématiques** supportent le type de données matrice. Si vous lancez un VI à partir de LabVIEW 7.x dans LabVIEW 8.x et que ce VI contient un VI mathématique câblé à une fonction qui peut utiliser le type de données matrice, un symbole rouge 7.x apparaît sur la fonction pour indiquer qu'elle utilise le comportement de LabVIEW 7.x.

## Fonctions de conversion de chaînes en nombres

Dans LabVIEW 7.x, les fonctions Nombre en chaîne hexadécimale, Nombre en chaîne octale et Nombre en chaîne décimale arrondissent une entrée flottante en un entier 32 bits avant de la convertir en chaîne.

Dans LabVIEW 8.x, ces fonctions arrondissent les entrées flottantes en entiers 64 bits avant de les convertir en chaînes. Cependant, si vous ouvrez un VI créé avec LabVIEW 7.x dans LabVIEW 8.x, LabVIEW maintient la compatibilité et la fonctionnalité en arrondissant les entrées flottantes en entiers 32 bits.

## Fonction Ouvrir une référence de VI

Dans LabVIEW 7.x, si l'entrée **chemin du VI** de la fonction Ouvrir une référence de VI est un chemin et qu'un VI en mémoire porte le même nom, LabVIEW renvoie une référence au VI en mémoire, même si la chemin du VI en mémoire ne correspond pas au chemin spécifié.

Dans LabVIEW 8.x, si l'entrée **chemin du VI** de la fonction Ouvrir une référence de VI est une chaîne, LabVIEW n'ouvre le VI que si **chemin du VI** correspond au nom qualifié complet d'un VI en mémoire sur cette cible. Si **chemin du VI** est un chemin, LabVIEW recherche un VI en mémoire avec le même chemin sur la même cible. S'il n'existe pas de VI avec un tel chemin, LabVIEW essaye alors de charger le VI à partir du disque, à l'emplacement spécifié. Une erreur survient si LabVIEW ne parvient pas à trouver le fichier, ou si le fichier crée un conflit avec un autre VI en mémoire ou des cibles.

## VI Mise à l'échelle rapide

Dans LabVIEW 7.1 et les versions antérieures, si l'entrée **X** du VI Mise à l'échelle rapide 1D ou 2D est un tableau de zéros, ce VI renvoie **0** pour **max|X|** et un tableau de NaN pour **Y[i]=X[i]/Max|X|** ou **Yij=Xij/Max|X|**. Dans LabVIEW 8.x, si l'entrée **X** du VI Mise à l'échelle rapide est un tableau de zéros, ce VI renvoie **0** pour **max|X|** et un tableau de zéros pour **Y[i]=X[i]/Max|X|** ou **Yij=Xij/Max|X|**.

## VI Lire une clé

Dans LabVIEW 7.x et versions précédentes, vous pouvez utiliser le VI Lire une clé pour lire une chaîne de caractères multi-octets codée en Shift-JIS. Vous devez câbler 1 ou <Shift-JIS> à l'entrée **encodage multi-octet**. Dans LabVIEW 8.x, par défaut, le VI Lire une clé lit les chaînes codées à caractères multi-octets si vous paramétrez les options régionales du système d'exploitation sur la méthode d'encodage appropriée.

## VI Mise à l'échelle

Dans LabVIEW 7.1 et les versions antérieures, si l'entrée **X** du VI Mise à l'échelle 1D ou 2D est un tableau de zéros, ce VI renvoie **0** pour **échelle**, **0** pour **offset** et un tableau de NaN pour **Y=(X\_offset)/échelle**. Dans LabVIEW 8.x, si l'entrée **X** du VI Mise à l'échelle est un tableau de zéros, ce VI renvoie **1** pour **échelle**, **0** pour **offset** et un tableau de zéros pour **Y=(X\_offset)/échelle**.

## Vis Sémaphore

Dans LabVIEW 7.x, le VI Libérer un sémaphore et le VI Acquérir un sémaphore ne tentent pas de s'exécuter si l'entrée **entrée d'erreur** reçoit une erreur. Dans LabVIEW 8.x, ces VI essaient de s'exécuter même si l'entrée **entrée d'erreur** reçoit une erreur. Cependant, si vous ouvrez un VI créé avec LabVIEW 7.x dans LabVIEW 8.x, LabVIEW maintient le comportement de LabVIEW 7.x.

## Vis d'e-mail SMTP

Dans LabVIEW 7.x et les versions précédentes, vous pouvez spécifier un jeu de caractères en câblant une valeur à l'entrée **jeu de caractères** des VIs d'e-mail SMTP. Dans LabVIEW 8.x, les VIs d'e-mail SMTP supposent que le message est codé avec le jeu de caractères (local) du système. Ces VIs codent le message au format UTF-8 avant d'envoyer l'e-mail. Les VIs E-mail SMTP n'utilisent plus les paramètres **jeu de caractères** ou **translit**.

## VI Trier des nombres complexes

Dans LabVIEW 7.x et versions antérieures, si vous paramétrez l'entrée **méthode** du VI Trier des nombres complexes à **Amplitude**, LabVIEW ne change pas la séquence des éléments ayant la même amplitude. Dans LabVIEW 8.x, si vous paramétrez **méthode** à **Amplitude**, LabVIEW classe les éléments de même amplitude d'abord en fonction de leurs parties réelles puis en fonction de leurs parties imaginaires.

## VI Vecteur unité

Dans LabVIEW 7.x et versions antérieures, le VI Vecteur unité calcule la norme du vecteur en entrée à l'aide de l'équation suivante :

$$\|X\| = \sqrt{x_0^2 + x_1^2 + \dots + x_{n-1}^2}$$

Dans LabVIEW 8.x, le VI Vecteur unité calcule la norme du vecteur en entrée en utilisant la formule suivante :

$$\|X\| = \left( |x_0|^y + |x_1|^y + \dots + |x_{n-1}|^y \right)^{\frac{1}{y}}$$

où  $X$  représente le vecteur en entrée,  $\|X\|$  représente la norme et  $y$  est le type de norme.

## VIs utilisateur

Les VIs que vous placez dans les répertoires `labview\help`, `labview\project` ou `labview\wizard` apparaissent respectivement dans les menus **Aide**, **Outils** et **Fichier**. Les VIs que vous placez dans ces répertoires dans LabVIEW 7.x et versions antérieures risquent de ne pas fonctionner de la manière attendue dans LabVIEW 8.x car LabVIEW 8.0 ou version ultérieure ouvre ces VIs dans une instance d'application privée.

Utilisez le VI `VIMemory Get VIs in Memory` dans le répertoire `labview\vi.lib\Utility\allVIsInMemory.llb` pour générer la liste de tous les VIs utilisateur en mémoire dans toutes les instances d'application. Utilisez le VI `Get User Application Reference` dans le répertoire `labview\vi.lib\Utility\allVIsInMemory.llb` pour créer une référence à l'instance d'application actuelle. Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur les instances d'application.

## VIs et fonctions obsolètes

LabVIEW 8.x ne supporte pas les fonctions et VIs suivants :

- LabVIEW 7.1 et les versions ultérieures n'installent pas le VI Compter les zéros d'un polynôme réel (Polynomial Real Zero Counter). Veuillez utiliser le VI Polynomial Real Zeros Counter à la place. Le titre français du VI est resté le même : Compter les zéros d'un polynôme réel.
- **(Mac OS)** LabVIEW 7.1 et les versions ultérieures n'installent pas les VIs PPC. Utilisez les VIs TCP à la place.
- LabVIEW 8.x ne supporte pas le VI Factorisation en QR. Utilisez le VI Décomposition en QR à la place.

- LabVIEW 8.x ne supporte pas les VIs Levenberg Marquardt et Ajustement non linéaire de Levenberg-Marquardt. Utilisez plutôt le VI Ajustement de courbe non linéaire.
- Dans LabVIEW 8.x, la fonction VISA Status Description ne se trouve plus sur la palette **Fonctions**. Utilisez plutôt les VIs Gestionnaire d'erreur simple ou Gestionnaire d'erreur général.
- LabVIEW 8.x ne supporte pas les VIs Distribution du khi-carré, Distribution F, Distribution normale et Distribution T. Utilisez respectivement les instances Khi carré, F, normale et t de Student du VI CDF continue.
- LabVIEW 8.x ne supporte pas les VIs Distribution du khi-carré inv., Distribution F inv., Distribution normale inv. et Distribution T inv. Utilisez respectivement les instances Khi carré, F, normale et t de Student du VI CDF continue inverse.
- Dans LabVIEW 8.x, les VIs Évaluation linéaire 1D et Évaluation linéaire 2D ne figurent plus sur la palette **Fonctions**. À la place, utilisez le VI Évaluation linéaire.
- Dans LabVIEW 8.x, les VIs Évaluation polynomiale 1D et Évaluation polynomiale 2D ne figurent plus sur la palette **Fonctions**. À la place, utilisez le VI Évaluation polynomiale.
- Dans LabVIEW 8.x, les VIs Cartésien en polaire 1D et Polaire en cartésien 1D ne figurent plus sur la palette **Fonctions**. À la place, utilisez les fonctions Re/Im en polaire et Polaire en Re/Im.
- Dans LabVIEW 8.x, le VI Analyseur harmonique ne figure plus sur la palette **Fonctions**. À la place, utilisez le VI Analyseur de distorsion harmonique si vous voulez mesurer les sorties **DHT** ou **amplitudes des composantes**, ou le VI Analyseur SINAD si vous voulez mesurer les sorties **SINAD** ou **DHT pus bruit**.
- Dans LabVIEW 8.x, le VI Fonctions de réseau (moy) ne figure plus sur la palette **Fonctions**. À la place, utilisez les VIs Fonction de réponse en fréquence (Amp-Phase), Fonction de réponse en fréquence (Réel-Im), Spectre croisé (Amp-Phase) ou Spectre croisé (Réel-Im).
- Dans LabVIEW 8.x, le VI Paramètres d'impulsion ne figure plus sur la palette **Fonctions**. À la place, utilisez le VI Mesures de transition pour mesurer les sorties **vitesse de variation**, **durée**, **overshoot** ou **preshoot**, le VI Mesures d'impulsion pour mesurer les sorties **période**, **durée de l'impulsion** ou **rapport cyclique**, ou le VI Amplitude et niveaux pour mesurer les sorties **amplitude**, **niveau de l'état haut** ou **niveau de l'état bas**.
- Dans LabVIEW 8.x, le VI Fonction de transfert ne se trouve plus sur la palette **Fonctions**. À la place, utilisez les VIs Fonction de réponse en fréquence (Amp-Phase) ou Fonction de réponse en fréquence (Réel-Im).

- Dans LabVIEW 8.x, le VI Express Assistant de rapport NI DIAdem ne figure plus sur la palette **Fonctions**. À la place, utilisez le VI Express Rapport DIAdem.
- Dans LabVIEW 8.x, la constante Nom de ressource VISA et la constante Nom logique IVI ne figurent plus sur la palette **Fonctions**. Pour spécifier un nom de ressource VISA, utilisez l'entrée **nom de ressource VISA** des VIs VISA. Pour spécifier un nom logique IVI, utilisez l'entrée appropriée du VI du driver qui initialise l'instrument.
- Dans LabVIEW 8.x, la constante menu déroulant d'erreurs ne se trouve plus sur la palette **Fonctions**. À la place, utilisez une constante entier signé 32 bits pour entrer le code d'erreur de votre choix.
- **(Windows et Linux)** Les VIs de son disponibles sur la palette **Son** dans LabVIEW 7.x ne figurent plus sur la palette **Fonctions** dans LabVIEW 8.x. À la place, utilisez les VIs de son dans LabVIEW 8.x. Les exemples fournis avec LabVIEW 7.x ne font plus partie de LabVIEW 8.x.
- **(Mac OS)** LabVIEW 8.x est toujours livré avec les VIs de son livrés avec LabVIEW 7.1. Les exemples fournis avec LabVIEW 7.x ne font plus partie de LabVIEW 8.x.

## VIs et fonctions d'E/S sur fichiers

Dans LabVIEW 8.x, le VI Lire des caractères dans un fichier ne se trouve plus sur la palette **Fonctions**. À la place, utilisez la fonction Lire un fichier texte.

Dans LabVIEW 8.x, le VI Ouvrir/Créer/Remplacer un fichier ne figure plus sur la palette **Fonctions**. Remplacez-le par la fonction Ouvrir/Créer/Remplacer un fichier. Les fonctions suivantes comprennent certaines des fonctionnalités du VI Ouvrir/Créer/Remplacer un fichier de LabVIEW 7.x et versions antérieures :

- Utilisez la fonction Obtenir la taille du fichier pour déterminer la taille d'un fichier.
- Utilisez le VI Express Boîte de dialogue de fichier pour spécifier le chemin de départ, le filtre de fichier et le nom par défaut d'un fichier ou d'un répertoire d'une boîte de dialogue.
- Utilisez la fonction Refnum en chemin pour convertir une référence en chemin.
- Utilisez la fonction Écrire dans un fichier binaire pour créer des fichiers texte indépendants de la plate-forme, ou d'autres types de fichiers binaires, et utilisez la fonction Lire un fichier binaire pour lire les fichiers binaires résultants.

Dans LabVIEW 8.x, les fonctions Lire un fichier et Écrire dans un fichier ne figurent plus sur la palette **Fonctions**. À la place, utilisez les fonctions Lire un fichier binaire et Écrire dans un fichier binaire.

Dans LabVIEW 8.x, le VI Écrire des caractères dans un fichier ne se trouve plus sur la palette **Fonctions**. À la place, utilisez la fonction Écrire dans un fichier texte.

Dans LabVIEW 8.x, la fonction Droits d'accès ne se trouve plus sur la palette **Fonctions**. À la place, utilisez les fonctions Obtenir les autorisations et Définir les autorisations.

Dans LabVIEW 8.x, la fonction EOF ne figure plus sur la palette **Fonctions**. À la place, utilisez les fonctions Obtenir la taille du fichier et Définir la taille du fichier.

Dans LabVIEW 8.x, la fonction Lister le contenu d'un répertoire ne se trouve plus sur la palette **Fonctions**. À la place, utilisez la fonction Lister le contenu du dossier.

Dans LabVIEW 8.x, la fonction Verrouiller la gamme ne se trouve plus sur la palette **Fonctions**. À la place, utilisez la fonction Interdire l'accès.

Si vous ouvrez un VI construit dans LabVIEW 7.x qui comporte la fonction Nouveau répertoire sur son diagramme, LabVIEW 8.x remplace cette fonction par la fonction Créer un dossier. Si le dossier que vous aviez spécifié dans l'entrée **chemin** n'existe pas, la fonction Créer un dossier crée le répertoire au lieu de renvoyer une erreur comme le faisait la fonction Nouveau répertoire.

Dans LabVIEW 8.x, la fonction Rechercher ne figure plus sur la palette **Fonctions**. À la place, utilisez les fonctions Obtenir la position dans le fichier et Définir la position dans le fichier.

Dans LabVIEW 8.x, la fonction Type et créateur ne se trouve plus sur la palette **Fonctions**. À la place, utilisez les fonctions Obtenir le type et le créateur et Définir le type et le créateur.

Dans LabVIEW 8.x, la fonction Infos sur le volume ne se trouve plus sur la palette **Fonctions**. À la place, utilisez la fonction Obtenir les infos sur le volume.

Dans LabVIEW 8.x, les fonctions Ouvrir un fichier et Nouveau un fichier ne figurent plus sur la palette **Fonctions**. Le VI Lire des lignes dans un fichier ne figure plus sur la palette **Fonctions** mais est livré avec LabVIEW pour des raisons de compatibilité.

Dans LabVIEW 8.x, les VIs Lire dans un fichier I16, Lire dans un fichier SGL, Écrire dans un fichier I16 et Écrire dans un fichier SGL ne se trouvent pas dans la palette **Fonctions**. À la place, utilisez les VIs Lire un fichier binaire et Écrire dans un fichier binaire.

## Modification du comportement de propriétés, méthodes et événements

Le comportement des propriétés, méthodes et événements suivants a changé dans LabVIEW 7.1 ou 8.0.

### Propriétés et méthodes d'Application

Dans LabVIEW 8.x, le comportement de certaines propriétés et méthodes de la classe Application dépend de l'instance d'application à laquelle elles appartiennent. Par exemple, le comportement de la propriété Application : VIs en mémoire dépend de l'instance d'application dans laquelle vous l'utilisez. Cette propriété renvoie la liste de tous les VIs en mémoire de l'instance d'application où se trouve la propriété. Cependant, le comportement de la propriété Application : chemin du répertoire ne dépend pas de l'instance d'application dans laquelle vous l'utilisez. Cette propriété renvoie le chemin absolu du répertoire dans lequel se trouve l'application. Cette information reste la même quelle que soit l'instance d'application.

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur les instances d'application.

### Méthode Face-avant : ouvrir

La méthode de LabVIEW 7.0 Ouvrir FA a été renommée Ancien Ouvrir FA dans LabVIEW 7.1. LabVIEW 7.1 intègre une méthode Ouvrir FA différente qui ne renvoie pas d'erreur si la face-avant est déjà ouverte. Dans LabVIEW 8.x, la méthode de LabVIEW 7.1 Ouvrir FA s'appelle désormais Face Avant : ouvrir. Si vous avez des VIs qui utilisent la méthode Ancien Ouvrir FA, veuillez remplacer la méthode avec la méthode Face Avant : ouvrir.

### Méthode Exécuter le VI

Dans LabVIEW 7.1, si vous définissez l'entrée **Supprimer ref. autom.** de la méthode Exécuter le VI à VRAI, LabVIEW supprime automatiquement la référence à la fin de l'exécution du VI. De plus, dans LabVIEW 8.x et versions ultérieures, LabVIEW supprime immédiatement la référence si la méthode renvoie une erreur. Ce comportement risque de casser un VI à l'exécution si une partie du diagramme dépend de cette référence.

## Événements Touche appuyée et Touche maintenue appuyée

Le champ de données **CléVirt** des événements Touche appuyée, Touche appuyée ?, Touche maintenue appuyée, et Touche maintenue appuyée ? des VIs et des commandes possède désormais des valeurs séparées pour les touches <Entrée> du pavé numérique et <Retour> du clavier alphanumérique. Dans LabVIEW 7.x et les versions précédentes, lorsque ces touches génèrent un de ces événements, LabVIEW renvoie **<Enter>** dans le champ de données **CléVirt**. Dans LabVIEW 8.x, lorsque ces touches génèrent un de ces événements, LabVIEW renvoie respectivement **<Enter>** ou **<Return>** dans le champ de données **CléVirt**.

**(Mac OS)** LabVIEW 8.x n'accepte que <Ctrl>-clic pour les menus locaux ; il n'accepte pas <Commande>-clic. Si vous émulez ce comportement avec une structure événement, vous devez modifier vos VIs pour émuler le nouveau comportement.

## Propriétés de ListeDéroulante

Dans LabVIEW 7x et versions antérieures, si vous attribuez la propriété Ligne du haut dans une liste déroulante à une ligne qui se trouve au-delà du dernier élément de cette liste, LabVIEW arrête le défilement à la dernière ligne visible de la liste. Dans LabVIEW 8.x, le nombre d'éléments visibles dans la liste déroulante ne limite pas le nombre de lignes que vous pouvez câbler à cette propriété.

LabVIEW 8.x ne supporte pas la propriété Double-clic pour les listes déroulantes à colonne unique. À la place, utilisez la méthode Obtenir la ligne double-cliquée.

## Propriété VI propriétaire

Dans LabVIEW 7.x et les versions antérieures, la propriété VI propriétaire renvoie une référence au VI auquel l'objet appartient. Cette référence garde le VI en mémoire. Dans LabVIEW 8.x, la référence renvoyée par la propriété VI propriétaire ne garde pas le VI en mémoire. Si le VI propriétaire est supprimé de la mémoire, cette référence devient non valide. Utilisez la fonction Ouvrir une référence de VI pour obtenir une référence à un VI qui reste en mémoire jusqu'à ce que vous fermiez cette référence de façon explicite.

## Propriété de Texte

Dans LabVIEW 7.x et versions antérieures, la propriété de Texte renvoie une chaîne en affichage normal. Dans LabVIEW 8.x, la propriété de Texte renvoie une chaîne dont le mode d'affichage du texte est identique à celui de l'objet de la face-avant. Par exemple, si vous affichez une commande chaîne en mode Affichage style mot de passe, la propriété de Texte renvoie la chaîne en mode Affichage style mot de passe.

## Propriétés de CommandeArbre

Dans LabVIEW 7.x et les versions antérieures, les propriétés Propriétés de la cellule active : taille de la cellule : hauteur et Propriétés de la cellule active : taille de la cellule : largeur renvoient 17 pixels pour chaque ligne de la commande arbre. Dans LabVIEW 8.x, les propriétés Cellule active : taille de la cellule : hauteur et Cellule active : taille de la cellule : largeur renvoient 16 pixels pour chaque ligne de la commande arbre.

## Méthode Chaînes de VI

Les chaînes que vous exportez des versions antérieures de LabVIEW en utilisant la méthode Exporter les chaînes du VI risquent de ne pas être importées correctement dans LabVIEW 8.x si vous utilisez la méthode Chaînes de VI : importer.

## Propriétés, méthodes et événements obsolètes

LabVIEW 8.x ne supporte pas les propriétés, les méthodes et les événements suivants :

### Propriétés de Curseur

LabVIEW 8.x ne supporte pas la propriété Style de verrouillage du curseur. À la place, utilisez la propriété Mode du curseur.

### Propriétés et événements de ListeDéroulante, Table, TableNumériques et CommandeArbre

LabVIEW 8.x ne supporte pas la propriété Couleur de premier plan de la cellule pour les listes multicolonnes. À la place, utilisez la propriété Cellule active : police de la cellule : couleur.

LabVIEW 8.x ne supporte pas la propriété Couleur de premier plan de cellule pour les tables ou les tables numériques. À la place, utilisez la propriété Cellule active : police de la cellule : couleur, pour les tables et les tables numériques.

LabVIEW 8.x ne supporte pas la propriété Cellule active : couleur de premier plan pour les commandes arbre. À la place, utilisez la propriété Cellule active : police de la cellule : couleur.

LabVIEW 8.x ne supporte plus les événements Glisser, Glisser ?, Déposer et Déposer ? pour la classe CommandeArbre. À la place, utilisez les événements Fin de glisser, Entrée de glisser, Sortie de glisser, Glisser sur, Mise à jour source de glisser, Début de glisser, Début de glisser ? et Déposer de la classe Commande.

## Propriétés de NumériqueNommé

LabVIEW 8.x ne prend pas en charge les propriétés Couleurs du numérique nommé, Couleurs du numérique nommé : couleur d'arrière-plan et Couleurs du numérique nommé : couleur du texte pour les objets numériques nommés. À la place, utilisez respectivement les propriétés Couleurs du texte, Couleurs du texte : couleur d'arrière-plan, et Couleurs du texte : couleur du texte.

## Propriétés de Face-avant

LabVIEW 8.x ne supporte plus la propriété Couleur pour la classe Face-avant. Si vous utilisez cette propriété dans LabVIEW 8.x, elle ne s'appliquera qu'au panneau supérieur de gauche. À la place, utilisez la propriété Couleur du panneau de la classe Panneau.

## Propriétés de Face-avantSecondaire

Dans LabVIEW 8.x, utilisez le panneau d'un sous-VI d'une face-avant secondaire pour configurer la visibilité des barres de défilement pour les commandes de la face-avant secondaire et mettre à l'échelle la face-avant dans les commandes de face-avant secondaire.

LabVIEW 8.x ne prend pas en charge la propriété Barre de défilement des X visible pour les commandes de face-avant secondaire. À la place, utilisez la propriété Visibilité de la barre de défilement horizontale pour les panneaux.

LabVIEW 8.x ne prend pas en charge la propriété Barre de défilement des Y visible pour les commandes de face-avant secondaire. À la place, utilisez la propriété Visibilité de la barre de défilement verticale pour les faces-avant.

LabVIEW 8.x ne prend pas en charge la propriété Mettre la face-avant à l'échelle pour les commandes de face-avant secondaire. À la place, utilisez la méthode Définir le mode de mise à l'échelle pour les faces-avant.

## Propriétés, méthodes et événements de VI

LabVIEW 8.x ne supporte pas la propriété Fenêtre face-avant : auto-centrer. À la place, utilisez la méthode Face-avant : centrer.

LabVIEW 8.x ne supporte pas la propriété Fenêtre face-avant : ajuster à l'écran. À la place, utilisez la propriété Fenêtre face-avant : état.

LabVIEW 8.x ne supporte plus la propriété Fenêtre face-avant : origine pour la classe VI. Si vous utilisez cette propriété dans LabVIEW 8.x, elle ne s'appliquera qu'au panneau supérieur de gauche. À la place, utilisez la propriété Origine de la classe Panneau.

LabVIEW 8.x ne supporte plus la propriété Fenêtre face-avant : afficher les barres de défilement pour la classe VI. Si vous utilisez cette propriété dans LabVIEW 8.x, elle ne s'appliquera qu'au panneau supérieur de gauche. À la place, utilisez les propriétés Visibilité de la barre de défilement horizontale et Visibilité de la barre de défilement verticale de la classe Panneau.

LabVIEW 8.x ne supporte plus les méthodes Obtenir le mode de mise à l'échelle de la face-avant et Définir le mode de mise à l'échelle de la face-avant pour la classe VI. Si vous utilisez ces méthodes dans LabVIEW 8.x, elles ne s'appliqueront qu'au panneau supérieur gauche. À la place, utilisez les méthodes Obtenir le mode de mise à l'échelle et Définir le mode de mise à l'échelle de la classe Panneau.

LabVIEW 8.x ne supporte plus les événements Clic de souris, Clic de souris ? Entrée de la souris, Sortie de la souris, Déplacement de la souris et Relâchement du bouton de souris pour la classe VI. À la place, utilisez respectivement les événements Souris appuyée, Souris appuyée ?, Entrée de la souris, Déplacement de la souris et Souris relâchée de la classe Panneau.

## Tags des éléments de l'application

Les tags d'éléments d'application suivants n'existent plus dans LabVIEW 8.x :

- APP\_BUILD\_STANDALONE\_APP
- APP\_DN\_ASSEMBLY\_REFS
- APP\_EDIT\_VI\_LIBRARY
- APP\_SAVE\_WITH\_OPTIONS
- APP\_SHOW\_CLIPBOARD
- APP\_SRC\_CODE\_CTRL
- APP\_SWITCH\_EXEC\_TARGET
- APP\_UPDATE\_VXI
- APP\_VIEW\_PRINTED\_MANUALS

Si vous utilisez un fichier menu d'exécution (.rtm) qui a été enregistré dans une version antérieure de LabVIEW et que ce fichier contient un tag qui n'existe plus, LabVIEW 8.x supprime automatiquement le tag du fichier .rtm lorsque vous enregistrez le fichier dans la boîte de dialogue **Éditeur de menu**. Les tags d'éléments d'application supprimés sont réservés par LabVIEW et ne peuvent pas être utilisés en tant que tags utilisateur.

## Support de HiQ

National Instruments ne prend pas en charge la fonctionnalité HiQ sous LabVIEW 8.x. Si une application utilise les VIs HiQ, vous pourriez les remplacer par les VIs de traitement du signal et de mathématiques.

## Fenêtre Liste des erreurs

Dans LabVIEW 7.x et versions antérieures, la section **Liste des VIs** de la fenêtre **Liste des erreurs** affiche les erreurs pour tous les VIs en mémoire. Dans LabVIEW 8.x, la section **Éléments avec erreurs** de la fenêtre **Liste des erreurs** affiche les erreurs pour tous les éléments en mémoire, tels que les VIs et les bibliothèques. Si au moins deux éléments ont le même nom, cette section indique l'instance d'application spécifique pour chaque élément ambigu. Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur les instances d'application.

## Syntaxe du fichier d'import/export des VIs

LabVIEW 8.x s'attend à trouver un nouveau jeu de tags, `<GROUPER></GROUPER>`, lorsque vous importez les chaînes d'un VI en sélectionnant **Outils>Avancé>Importer les chaînes** ou en utilisant la méthode Chaînes de VI : importer. Ce nouveau tag représente les objets de la face-avant qui sont groupés les uns aux autres. En conséquence, sous LabVIEW 8.x, vous ne pouvez pas importer des fichiers de chaînes de VI créés dans une version antérieure de LabVIEW.

Dans LabVIEW 7.1 et versions antérieures, les chaînes d'une liste déroulante sont codées dans la section `<ITEMS>` de ses données privées. Dans LabVIEW 8.x, les chaînes d'une liste déroulante sont répertoriées dans la section `<STRINGS>` de ses données privées. En outre, dans LabVIEW 7.1 et versions antérieures, une liste déroulante ne peut avoir qu'une police qui s'affiche dans la section `<LBLABEL>` de ses données privées. Dans LabVIEW 8.x, une liste déroulante peut avoir plusieurs polices, répertoriées dans la section `<CELL_FONTS>` de ses données privées.

Dans LabVIEW 7.1 et versions antérieures, les chaînes d'une liste multicolonne sont codées dans la section des données par défaut. Toutefois, les données par défaut dans une liste multicolonne peuvent être un entier ou un tableau d'entiers. Dans LabVIEW 8.x, les chaînes d'une liste multicolonne sont répertoriées dans la section de ses données privées.

LabVIEW 7.1 et les versions antérieures n'exportent ni les chaînes ni les polices d'une commande arbre. LabVIEW 8.x peut exporter à la fois les chaînes et les polices d'un arbre, et ce, dans le même format que celui utilisé pour les listes déroulantes et les listes multicolennes.

Dans LabVIEW 8.x, chaque ligne d'un fichier d'exportation contient au maximum deux tags pour les données par défaut ou privées. LabVIEW 8.x indente aussi les éléments une fois à chaque niveau d'imbrication.

Effectuez les étapes suivantes pour convertir des fichiers de chaînes de VI au format LabVIEW 8.x.

1. Importez le fichier de chaîne dans une version précédente de LabVIEW.
2. Enregistrez le VI.
3. Chargez le VI dans LabVIEW 8.x.
4. Sélectionnez **Outils**»**Avancé**»**Exporter les chaînes** afin de générer un fichier de chaînes de VI au format LabVIEW 8.x.

## Conversion des données du descripteur de type vers et à partir de LabVIEW 7.x

Le format dans lequel LabVIEW enregistre les descripteurs de type a changé dans LabVIEW 8.x. LabVIEW 7.x enregistre les descripteurs de type en représentation plate à 16 bits. LabVIEW 8.x enregistre les descripteurs de type en représentation plate à 32 bits. Ce changement élimine la limitation de taille à 64 Ko des descripteurs de type.

LabVIEW 8.x offre un mécanisme de lecture des descripteurs de type écrits sous LabVIEW 7.x et d'écriture de ces descripteurs de type dans un format lisible par LabVIEW 7.x. La fonction Aplatis en chaîne contient un élément de menu local **Convertir des données de la version 7.x**. Si vous effectuez un clic droit et sélectionnez cet élément, la fonction traite les données d'entrée comme si elles avaient été écrites pour LabVIEW 7.x. Si vous sélectionnez l'élément de menu local **Convertir des données de la version 7.x** alors que la sortie **chaîne de données** est câblée, LabVIEW 8.x trace un 7.x rouge sur la fonction pour indiquer qu'elle convertit des données vers ou depuis LabVIEW 7.x. Pour éviter la conversion des données, sélectionnez à nouveau l'élément de menu local **Convertir des données de la version 7.x** pour décocher la case.

Dans LabVIEW 8.x, lorsque vous chargez un VI enregistré avec LabVIEW 7.x ou une version antérieure, LabVIEW 8.x définit automatiquement l'attribut **Convertir des données de la version 7.x** sur la fonction Aplatis en chaîne. La fonction continue à fonctionner comme dans LabVIEW 7.x et les versions antérieures. Si vous souhaitez qu'un VI utilise le format de descripteur de type de LabVIEW 8.x, faites un clic droit sur la fonction Aplatis en chaîne et sélectionnez **Convertir des données de la version 7.x** dans le menu local pour supprimer la marque de sélection. Utilisez le format de descripteur de type de LabVIEW 8.x si les VIs n'ont pas besoin de manipuler des fichiers qui contiennent des données écrites

dans LabVIEW 7.x ou version antérieure, et qu'ils n'envoient ni ne reçoivent de données en direction ou en provenance de VIs qui s'exécutent sous LabVIEW 7.x ou version antérieure. Le format précédent des descripteurs de types ne sera peut-être plus supporté dans les versions futures de LabVIEW.

## Migration à partir du gestionnaire de contrôle de code source intégré à LabVIEW

Le gestionnaire de contrôle de source intégré de LabVIEW 7.x et des versions antérieures n'est pas disponible dans LabVIEW 8.x. Si vous voulez utiliser le contrôle de source dans LabVIEW, vous devez sélectionner un gestionnaire de contrôle source fourni par un tiers. Si vous utilisiez le gestionnaire intégré dans les versions antérieures, vous devez migrer les fichiers vers un autre gestionnaire afin de pouvoir utiliser le contrôle de source dans LabVIEW. Reportez-vous au site Web de National Instruments à l'adresse [ni.com/frinfo](http://ni.com/frinfo) et entrez l'info-code `exgucn` pour obtenir la toute dernière liste des gestionnaires de contrôle de source de tiers supportés par LabVIEW.

Quand vous migrez les fichiers vers un nouveau gestionnaire de contrôle de source, vous perdez l'historique des révisions qui était enregistré dans le gestionnaire intégré. Vous ne pouvez pas transférer les versions précédentes des fichiers dans le nouveau gestionnaire.

Suivez les étapes ci-après pour migrer les fichiers du gestionnaire de contrôle de source intégré vers un gestionnaire de contrôle de source de tiers.

1. Dans la version antérieure de LabVIEW, vérifiez que les fichiers inclus dans le gestionnaire de contrôle de source intégré de LabVIEW ont bien été soumis par les utilisateurs.
2. Sur l'ordinateur auquel vous voulez ajouter les fichiers au nouveau gestionnaire de contrôle de source, utilisez le gestionnaire intégré pour obtenir la version la plus récente de tous les fichiers.
3. Utilisez le gestionnaire pour obtenir les fichiers du contrôle de source.
4. À partir du gestionnaire de contrôle de source de tiers, configurez les paramètres que vous désirez pour le nouveau projet de contrôle de source.
5. Configurez LabVIEW pour qu'il fonctionne avec le gestionnaire de contrôle de source de tiers. Reportez-vous au livre **Principes de base» Organisation et gestion d'un projet» Procédures» Utilisation du contrôle de source dans LabVIEW** dans l'onglet **Sommaire** de l'*Aide LabVIEW* pour savoir comment configurer LabVIEW en vue de l'utilisation d'un gestionnaire de contrôle de source d'un tiers.

6. Créez un projet LabVIEW. Ajoutez les fichiers inclus dans le gestionnaire de contrôle de source intégré au projet. Quand LabVIEW vous y invite, ajoutez les fichiers au contrôle de source. Vous pouvez aussi ajouter les fichiers directement dans le gestionnaire de contrôle de source de tiers. Reportez-vous au livre **Principes de base» Organisation et gestion d'un projet» Procédures» Création d'un projet LabVIEW** dans l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations sur la création d'un projet LabVIEW.

## Conversion des chaînes NaN en types entiers (Windows)

Dans LabVIEW 7.x, lorsque vous convertissez explicitement ou implicitement un NaN en entier, la valeur devient la plus petite valeur pour le type de données de cet entier. Par exemple, la conversion de NaN en un entier signé 16 bits produit la valeur `_32768`, soit la plus petite valeur possible pour un entier signé 16 bits.

Dans LabVIEW 8.x, lorsque vous convertissez explicitement ou implicitement un NaN en entier, la valeur devient la plus grande valeur pour le type de données de cet entier. Par exemple, la conversion de NaN en un entier signé 16 bits produit la valeur `32767`, soit la plus grande valeur possible pour un entier signé 16 bits.

## Constantes câblées aux structures Condition

Dans LabVIEW 7.x, vous pouvez garder les sous-VIs en mémoire en câblant une constante à la structure Condition et en plaçant le sous-VI dans une condition non exécutée. Par exemple, si vous câblez une constante VRAI à la structure Condition et que vous placez un sous-VI dans le cas FAUX de cette structure, LabVIEW charge le sous-VI en même temps que le VI qui l'appelle. LabVIEW 8.x supprime tout code qui ne s'exécute pas. Ainsi, si vous chargez un VI dans LabVIEW 8.x qui était enregistré dans une version précédente de LabVIEW avec une constante câblée à une structure Condition, LabVIEW change la constante en commande masquée pour maintenir le comportement attendu dans les anciennes versions de LabVIEW.

## Report des messages du système d'exploitation

Dans LabVIEW 7.x, LabVIEW traite les messages du système d'exploitation tout en exécutant les VIs Callback pour gérer les événements .NET et ActiveX. Dans LabVIEW 8.x, LabVIEW reporte le traitement des messages du système d'exploitation jusqu'à ce que le VI Callback arrête de

s'exécuter ou que vous chargiez une boîte de dialogue modale. Ceci permet aux VIs Callback de s'exécuter sans interruption et empêche LabVIEW de générer un événement à l'intérieur d'un autre événement, ce qui pourrait mener à une impasse.

Vous ne pouvez pas effectuer d'appels synchrones vers des boîtes de dialogue non modales à partir d'un VI Callback. Vous devez appeler une boîte de dialogue non modale de manière asynchrone à partir d'un VI Callback en appelant une méthode Exécuter le VI sur la boîte de dialogue et en câblant une constante booléenne FAUX à l'entrée **Attendre la fin** de la méthode.

Dans LabVIEW 7.x, LabVIEW traite les messages du système d'exploitation tout en exécutant les fonctions des bibliothèques partagées ou des DLL. Dans LabVIEW 8.x, LabVIEW reporte le traitement des messages du système d'exploitation jusqu'à ce que les appels des fonctions de DLL soient terminés ou que vous chargiez une boîte de dialogue modale à partir de la DLL. Ceci permet aux fonctions de DLL de s'exécuter sans interruption et empêche LabVIEW d'appeler la même DLL lorsqu'une de ses fonctions est en cours d'exécution, ce qui risquerait de mener à une impasse.

Lorsque vous utilisez ce comportement par défaut, vous ne pouvez pas faire d'appels synchrones vers les boîtes de dialogues non modales si une DLL est en cours d'exécution. Vous devez appeler une boîte de dialogue non modale de manière asynchrone à partir d'une DLL en appelant une méthode Exécuter le VI sur la boîte de dialogue et en câblant une constante booléenne FAUX à l'entrée **Attendre la fin** de la méthode.

Vous pouvez reporter les messages du système d'exploitation dans les DLL que vous construisez. Cliquez avec le bouton droit sur la DLL dans la fenêtre de l'**Explorateur de projet**, sélectionnez **Propriétés** dans le menu local, sélectionnez **Avancé** dans la liste **Catégorie** et décochez l'option **Retarder les messages du système d'exploitation dans la bibliothèque partagée** pour traiter les messages du système d'exploitation pendant que les fonctions de la DLL s'exécutent.

## Resource Manager (Mac OS)

LabVIEW 7.x et versions antérieures offrent des capacités non documentées grâce auxquelles vous pouvez lire et écrire des fichiers ressource Macintosh. Dans LabVIEW 8.x, ces méthodes n'existent pas. Les utilitaires qui utilisent ces capacités non documentées ne fonctionnent pas ; vous ne pouvez donc plus lire ou écrire des fichiers ressource Macintosh à partir des VIs.

## Boîtes de dialogue à un et deux boutons

Dans LabVIEW 7.x et versions antérieures, vous ne pouvez pas arrêter l'exécution par programmation d'un VI qui possède une boîte de dialogue à un ou deux boutons. Dans LabVIEW 8.x, vous pouvez arrêter par programmation l'exécution d'un VI qui affiche ces boîtes de dialogue en utilisant la méthode de VI `Abandonner`.

## Nœuds de propriété et de méthode

Si vous créez un nœud de propriété ou de méthode lié implicitement à partir de la légende d'un curseur dans LabVIEW 7.x, LabVIEW supprime le nœud lorsque vous ouvrez le VI dans LabVIEW 8.x.

## Mise à jour des bibliothèques partagées

Si vous construisez une bibliothèque partagée (DLL) liée à `labview.lib` dans LabVIEW 7.x ou une version antérieure, liez-la à `labviewv.lib` dans LabVIEW 8.x. Reportez-vous à l'*Aide LabVIEW* pour savoir comment lier des bibliothèques partagées à `labviewv.lib`.

## Valeurs des marges d'impression

Dans LabVIEW 7.x et les versions antérieures, l'option **Marges** de la page **Impression** de la boîte de dialogue **Avancé** affiche des valeurs en centimètres. Dans LabVIEW 8.x, l'option **Marges** utilise des millimètres.

## Mise à jour à partir de LabVIEW 6.x

Il est possible que vous rencontriez les problèmes de compatibilité suivants lorsque vous passez de LabVIEW 6.x à LabVIEW 8.2. Reportez-vous aux sections [Mise à jour à partir de LabVIEW 7.x](#) et [Mise à jour à partir de LabVIEW 8.0](#) de ce document pour obtenir des informations sur d'autres problèmes de mise à jour que vous pourriez éventuellement rencontrer.

Reportez-vous aux *Notes de mise à jour LabVIEW* spécifiques à chaque version de LabVIEW entre les versions 6.x et 8.0, sur [ni.com/manuals](http://ni.com/manuals), pour obtenir des informations complémentaires sur les nouvelles fonctionnalités et modifications apportées à ces versions.

## Modifications apportées aux types de données waveform

Dans LabVIEW 7.0, les données de type waveform utilisent le type de données d'horodatage pour le composant **t0** au lieu d'un nombre à virgule flottante double précision. Si vous enregistrez des données de type waveform dans un fichier sans inclure d'informations sur le type de données dans LabVIEW 6.x, vous risquez d'obtenir une erreur si vous tentez de récupérer ces données dans LabVIEW 7.x et versions ultérieures.

Dans LabVIEW 7.x et les versions ultérieures, le VI Lire une waveform à partir d'un fichier convertit l'ancien format de type de données waveform dans un fichier au nouveau format de type de données waveform. Ce VI affiche une boîte de dialogue qui vous propose d'accepter la conversion. Dans le moteur d'exécution de LabVIEW, le VI Lire une waveform à partir d'un fichier ne peut pas effectuer cette conversion et de ce fait, renvoie une erreur. Reportez-vous au site Web de National Instruments [ni.com/frinfo](http://ni.com/frinfo) puis entrez l'info-code `exd9zq` pour en savoir plus sur la migration de données waveform de LabVIEW 6.x vers LabVIEW 7.x et versions ultérieures.

## VIs de compatibilité série

Dans LabVIEW 7.x et les versions ultérieures, les VIs de compatibilité série ne figurent pas sur la palette **Fonctions**. Utilisez les VIs et les fonctions VISA pour créer des VIs qui communiquent avec les périphériques VXI.

Dans LabVIEW 7.x et les versions ultérieures, LabVIEW n'utilise pas le driver `serpdrv` pour communiquer avec le driver série du système d'exploitation. LabVIEW comprend des VIs compatibles basés sur VISA. Pour les nouvelles applications, utilisez les VIs et les fonctions VISA et série afin de commander des périphériques série. Tous les VIs créés dans des versions antérieures de LabVIEW intégrant des VIs série fonctionnent toujours sous LabVIEW 7.1 et versions ultérieures.

Si vous avez reconfiguré le mappage des numéros assignés aux ports, vous devez spécifier le mappage de ces ports. Utilisez le VI Donner des noms aux ports série dans le répertoire `labview\vi.lib\Instr\_ser_sup.lib` pour spécifier le mappage des ports série. Câblez un tableau de chaînes à l'entrée **Alias VISA** du VI et entrez les noms de ports que vous utilisez dans le tableau d'entrée. Chaque élément du tableau doit correspondre à un port. Par exemple, si vous avez configuré le port 0 pour qu'il mappe la ressource VISA appelée MySerialPort, entrez `MySerialPort` comme premier élément du tableau d'entrée **Alias VISA**. Vous devez appeler le VI Donner des noms aux ports série avant d'appeler le VI VISA Configure Serial Port.

Reportez-vous à `labview\examples\instr\smplser1.llb` pour consulter des exemples d'utilisation des VIs et des fonctions VISA pour contrôler des instruments série.

## Données par défaut dans des boucles

Dans LabVIEW 6.0 et les versions antérieures, les boucles For produisent des données non définies si la boucle ne s'exécute pas. Dans LabVIEW 6.1 et les versions ultérieures, les boucles For produisent des valeurs par défaut si vous câblez 0 au terminal de décompte de la boucle For ou si vous câblez un tableau vide comme entrée à la boucle For alors que l'auto-indexation est activée. La boucle ne s'exécute pas, et toutes les sorties de tunnels pour lesquelles l'auto-indexation est désactivée contiennent la valeur par défaut pour le type de données du tunnel.

## Licence pour face-avant distante

Le système de développement complet LabVIEW (Full Development System) et l'Application Builder comprennent une licence de face-avant distante qui permet à un client d'afficher et de contrôler une face-avant à distance. Le système de développement professionnel LabVIEW comprend une licence de face-avant qui permet à cinq clients d'afficher et de contrôler une face-avant à distance.

Vous pouvez mettre à jour la licence de face-avant distante afin de supporter un plus grand nombre de clients.

## Allocation de threads multiples

LabVIEW 7.1 et les versions ultérieures allouent un plus grand nombre de threads pour l'exécution des VIs que les versions antérieures à LabVIEW 7.1. En raison de ce changement, des erreurs risquent de se produire avec des threads multiples si vous désignez de manière incorrecte les nœuds Appeler une fonction d'une DLL comme étant réentrants alors que la DLL que vous appelez n'est pas vraiment réentrante. Pour plus d'informations sur la fonction Appeler une fonction d'une DLL et la réentrance, consultez l'*Aide LabVIEW*.

Pour modifier la façon dont LabVIEW alloue des threads, utilisez le VI `threadconfig` dans `labview\vi.lib\Utility\sysinfo.llb`. Vous pouvez aussi désactiver la réentrance d'un VI en sélectionnant **Fichier» Propriétés du VI**, puis **Exécution** dans le menu déroulant **Catégorie** et en désélectionnant l'option **Exécution réentrante**.

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur l'allocation des threads.

## Drivers d'instruments

Le progiciel LabVIEW de LabVIEW 7.x et des versions ultérieures ne comprend pas le CD de la bibliothèque de drivers d'instruments LabVIEW, qui contient les drivers d'instruments. Téléchargez les drivers d'instruments de la base de drivers d'instruments NI à l'adresse [ni.com/idnet](http://ni.com/idnet). À la place, la boîte du progiciel LabVIEW contient un CD de drivers de périphériques National Instruments comprenant les drivers pour NI-DAQ, NI-VISA et d'autres drivers National Instruments.

## Unités et facteurs de conversion

Dans LabVIEW 7.x et les versions ultérieures, vous n'avez pas besoin d'utiliser la fonction Convertir une unité pour supprimer l'unité superflue après avoir utilisé la fonction Opérateur arithmétique.

Les facteurs de conversion d'unités de LabVIEW 7.1 et versions ultérieures se conforment plus étroitement aux directives publiées par le National Institute for Standards and Technology, ou NIST (Institut national des normes et de la technologie aux États-Unis), dans le document *Guide for the Use of the International System of Units (SI)*. En outre, l'unité `calorie` est désormais `calorie (thermique)` et `puissance` est désormais `puissance (électrique)`. Les abréviations utilisées pour ces unités n'ont pas changé. Le tableau suivant dresse la liste détaillée des modifications apportées aux facteurs de conversion d'unités entre LabVIEW 6.1 et 7.x et les versions ultérieures.

Unité	Définition dans la version 6.1	Définition dans les versions 7.x et ultérieures
unité astronomique (UA)	149 498 845 000 m	149,597,900,000 m
unité thermique anglaise (moyenne)	1055,79 J	1055,87 J
électron volt (eV)	1,602e <sub>-19</sub> J	1,60217642e <sub>-19</sub> J
bougie	10,764 lx	10,7639 lx
puissance en CV vs puissance (électrique)	745,7 W	746 W. La nouvelle conversion est exacte.
gallon - impérial	4,54596 l	4,54609 l
année lumière	9,4605 Pm	9,46073 Pm
livre force	4,448 N	4,448222 N
rod	16,5 ft	5,029210 m

Unité	Définition dans la version 6.1	Définition dans les versions 7.x et ultérieures
slug	32,174 lb	14,59390 kg
unité de masse atomique unifiée (u)	1,66057e_27 kg	1,66053873e_27 kg

## Propriété Retarder MàJ face-avant

Dans LabVIEW 6.1 et les versions antérieures, LabVIEW attendait que la propriété Retarder MàJ face-avant soit définie à FAUX avant de retracer les objets de la face-avant pour implémenter les changements apportés. Dans LabVIEW 7.0, quand vous définissez cette propriété à VRAI, LabVIEW retrace tous les objets de la face-avant en cours de modification, puis reporte toutes les nouvelles demandes de mise à jour de la face-avant. Dans certains cas, ceci pousse LabVIEW à retracer une fois de trop les objets modifiés de la face-avant.

## Gammes des données des commandes numériques

Dans LabVIEW 6.1 et les versions antérieures, certaines commandes numériques ont les valeurs par défaut suivantes : valeur minimale de 0,00, valeur maximale de 0,00, valeur d'incrément de 0,00 et action hors gamme **Ignorer**. Dans LabVIEW 7.x et versions ultérieures, ces commandes numériques utilisent les valeurs de gamme de données par défaut pour le type de données.

## Définitions de type et points de coercition

Dans LabVIEW 6.1 et les versions ultérieures, les fils de liaison comprennent des informations sur les définitions de type, c'est pourquoi vous constaterez probablement qu'il y a davantage de points de coercition sur les diagrammes. Si vous câblez une définition de type à un terminal de VI ou de fonction qui ne l'est pas, un point de coercition apparaît. Un point de coercition apparaît aussi si vous câblez un terminal de sortie qui est une définition de type à un indicateur qui ne l'est pas. Ces points de coercition vous indiquent l'endroit dans vos VIs où vous n'utilisez pas de façon cohérente les définitions de type. Dans ce cas, les points de coercition n'affectent pas la performance à l'exécution.

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations concernant l'utilisation de la fonction Aplatis en chaîne pour aplatis des définitions de type.

## Étiquette du bouton de la boîte de dialogue de fichier

Dans LabVIEW 6.1 et les versions antérieures, la boîte de dialogue qu'affiche la fonction Boîte de dialogue de fichier comporte l'étiquette de bouton **Enregistrer** si l'utilisateur a la possibilité d'entrer un nouveau nom de fichier. Sinon, l'étiquette de bouton est **Ouvrir**. Dans LabVIEW 8.x, l'étiquette du bouton de la boîte de dialogue de fichier affichée par le VI Express Boîte de dialogue de fichier est **OK** dans tous les cas, à moins que vous ne la changiez. Utilisez l'entrée **étiquette du bouton** du VI Express Boîte de dialogue de fichier pour modifier l'étiquette du bouton. Si vous utilisez le VI Express Boîte de dialogue de fichier dans un VI existant, envisagez d'évaluer le comportement du VI pour vous assurer que l'étiquette par défaut, **OK**, correspond bien à la fonctionnalité du VI.

## Fonction Contrôle de l'aide en ligne

L'entrée **Chemin du fichier d'aide** de la fonction Contrôle de l'aide en ligne est désormais requise. Vous pouvez câbler à cette entrée soit le nom du fichier d'aide compilé (.chm ou .hlp), soit le chemin complet du fichier d'aide compilé. Si vous câblez seulement un nom de fichier d'aide compilé, LabVIEW recherche ce fichier dans le répertoire `labview\help`.

## Affichage de la face-avant au chargement

Dans LabVIEW 7.x et versions ultérieures, si vous configurez un VI pour qu'il affiche sa face-avant lorsque LabVIEW le charge et que vous chargez ce VI à l'aide du VI Serveur, LabVIEW n'affiche pas la face-avant. Vous devez alors utiliser la méthode Face Avant : ouvrir pour afficher la face-avant par programmation.

## Fonction Ouvrir une référence de VI

Dans LabVIEW 6.1 et versions antérieures, si vous ne câblez pas une valeur au paramètre **options** de la fonction Ouvrir une référence de VI, LabVIEW crée une instance de VI à partir d'un modèle si le modèle n'est pas déjà en mémoire. Si le VI est en mémoire, LabVIEW ouvre une référence au modèle. Dans LabVIEW 7.0 et 7.1, si vous utilisez la fonction Ouvrir une référence de VI pour créer une référence à un modèle qui est déjà en mémoire, la fonction renvoie une erreur, sauf si vous spécifiez `0x02` dans le paramètre **options**. Dans LabVIEW 8.0 et version ultérieure, si vous utilisez la fonction Ouvrir une référence de VI pour créer une référence à un modèle, LabVIEW crée une instance de VI à partir du modèle même si ce modèle est déjà en mémoire.

## Représentation exponentielle

Dans LabVIEW 6.0 et versions antérieures, l'opérateur  $\wedge$  représente l'élévation à une puissance dans la boîte de calcul. Dans LabVIEW 6.1 et versions ultérieures, l'opérateur d'élévation à la puissance est  $**$ , par exemple :  $x**y$ . L'opérateur  $\wedge$  représente l'opération OU exclusif binaire ou (XOR).

## Fichier de stockage de la configuration IVI

Le format Fichier de stockage de la configuration IVI exige désormais que tous les noms soient sensibles à la casse. Si vous utilisez des noms logiques, des noms de session de driver ou des noms virtuels dans votre application, veillez à ce que le nom que vous utilisez corresponde exactement au nom défini dans le fichier de stockage de la configuration IVI, sans changer d'aucune manière la casse des caractères qui composent le nom.

## Formulaire de support technique

Dans LabVIEW 7.x et les versions ultérieures, le programme d'installation de LabVIEW n'installe pas `techsup.11b`. Reportez-vous à la page [ni.com/france/support](http://ni.com/france/support) du site Web de National Instruments pour trouver une réponse à vos questions et résoudre les problèmes liés à l'installation, la configuration et aux applications.

## Mise à jour à partir de LabVIEW 5.x ou versions antérieures

Reportez-vous au site Web de National Instruments à [ni.com/frinfo](http://ni.com/frinfo) et entrez l'info-code `ext8h9` pour savoir comment effectuer une mise à niveau de LabVIEW 5.x ou version antérieure, à LabVIEW 8.2.

## Fonctionnalités et changements dans LabVIEW 8.2

---

Reportez-vous à l'*Aide LabVIEW* pour obtenir davantage d'informations sur les concepts de programmation, consulter des instructions pas à pas et des informations de référence sur les fonctionnalités de LabVIEW 8.2. Accédez à l'*Aide LabVIEW* en sélectionnant **Aide»Rechercher dans l'Aide LabVIEW**.

Reportez-vous au fichier `readme.html` du répertoire `labview` pour obtenir des informations plus détaillées sur les problèmes connus, les problèmes de compatibilité et pour obtenir des informations sur des fonctionnalités ajoutées tardivement dans LabVIEW 8.2.

# Documentation de LabVIEW

Les améliorations suivantes ont été apportées à la documentation de LabVIEW 8.2 :

- Dans la version anglaise, un lien permettant de soumettre un commentaire apparaît en bas de toutes les rubriques de l'**Aide LabVIEW**. Pour soumettre vos commentaires sur une rubrique de l'aide, cliquez sur ce lien et remplissez le formulaire **Documentation Suggestion Details**. Ce lien n'apparaît que sur les versions anglaises de l'*Aide LabVIEW*.
- Les rubriques portant sur les concepts possèdent une table de navigation dans l'angle supérieur droit de la rubrique d'aide. Cliquez sur un lien dans le tableau pour passer directement à la sous-rubrique sélectionnée.

## Nouveaux VIs d'exemple

Reportez-vous au dossier **Nouveaux exemples pour LabVIEW 8.x** sur l'onglet **Parcourir** de l'Outil de recherche d'exemples NI pour lancer les VIs d'exemple qui ont été ajoutés à LabVIEW 8.x et pour obtenir leur description.

## Améliorations apportées au temps de lancement

LabVIEW 8.2 se lance plus rapidement que LabVIEW 8.0 grâce à l'optimisation des performances.

## Améliorations apportées au diagramme

Le diagramme de LabVIEW 8.2 et les fonctionnalités associées présentent les améliorations suivantes.

### Modifications des couleurs par défaut

Les couleurs par défaut des composants suivants du diagramme ont changé afin de mieux les distinguer :

- Les fils de liaison et les terminaux des clusters d'erreur apparaissent désormais en jaune et non en rose sur le diagramme.
- Par défaut, les points de coercition apparaissent désormais en rouge, plus en gris. Pour changer la couleur des points de coercition, sélectionnez **Outils»Options** puis **Couleurs** dans la liste **Catégorie**. Désélectionnez l'option **Utiliser les couleurs par défaut** et cliquez sur la boîte de couleur **Points de coercition** pour sélectionner une couleur différente.

## Suppression des points d'arrêt dans la hiérarchie d'un VI

Dans un VI, sélectionnez l'option **Édition**»**Supprimer les points d'arrêt de la hiérarchie** pour supprimer tous les points d'arrêt de la hiérarchie du VI. Vous devez supprimer manuellement les points d'arrêt dans les VIs appelés dynamiquement et les VIs référencés par la fonction Référence de VI statique.

## Optimisation des performances avec les constantes

LabVIEW utilise le constant folding pour optimiser les performances des VIs. Grâce au constant folding, LabVIEW stocke les valeurs des constantes lorsqu'il compile des VIs plutôt que de les calculer à l'exécution. Pour les constantes câblées à des structures, LabVIEW calcule les valeurs en sortie des structures quand il compile des VIs et stocke ces valeurs pour qu'elles soient disponibles au moment de l'exécution.

Vous pouvez afficher les hachures du constant folding sur le diagramme en sélectionnant **Outils**»**Options**, puis **Diagramme** dans la liste de **Catégorie**, et en cochant les cases de **Afficher le 'constant folding' des fils de liaison** et **Afficher le 'constant folding' des structures**. Lorsque vous sélectionnez l'option **Afficher le 'constant folding' des fils de liaison**, des hachures grises apparaissent sur les fils de liaison connectés aux constantes pour lesquelles le constant folding est activé. Lorsque vous sélectionnez l'option **Afficher le 'constant folding' des structures**, des hachures grises apparaissent dans les structures qui sont câblées à des constantes. Il se peut que les hachures n'apparaissent qu'une fois le VI exécuté ou enregistré.

LabVIEW 8.2 conserve également les constantes calculées que vous câblez aux terminaux de sélection des structures Condition et des terminaux de condition des boucles While.

## Autres améliorations apportées au diagramme

Le diagramme de LabVIEW 8.2 contient les autres améliorations suivantes :

- Lorsque vous cliquez avec le bouton droit sur la sortie **référence de sortie** d'une propriété ou d'un nœud de méthode et que vous sélectionnez **Créer** dans le menu local, le menu **Créer** affiche les propriétés ou les méthodes dans la même classe que le nœud de méthode ou de propriété.
- La propriété **Éléments visibles** : lignes de hiérarchie visibles accepte trois valeurs. Si elle est définie à **Visible**, LabVIEW affiche toujours les lignes pour indiquer la hiérarchie des éléments sous forme de lignes horizontales et verticales à gauche des éléments dans l'arborescence.

Si cette propriété est définie à **Invisible**, les lignes de la hiérarchie sont toujours invisibles. Si cette propriété est définie à **OS Default**, LabVIEW affiche les lignes de hiérarchie si les arborescences du système d'exploitation les affichent.

## Améliorations apportées à la face-avant

La face-avant de LabVIEW 8.2 et ses fonctionnalités associées présentent les améliorations suivantes.

### Définition d'images d'arrière-plan pour des panneaux

Vous pouvez définir et importer des images d'arrière-plan pour des panneaux. Cliquez avec le bouton droit sur la barre de défilement d'un panneau et sélectionnez **Propriétés** dans le menu local. Dans la boîte de dialogue **Propriétés du panneau**, sélectionnez une image dans la liste d'**Arrière-plan**.

Pour sélectionner une image qui ne figure pas dans la liste d'**Arrière-plan**, cliquez sur le bouton **Parcourir**. LabVIEW supporte les formats graphiques BMP, JPEG et PNG pour les images d'arrière-plan. Vous pouvez aussi utiliser la propriété Image d'arrière-plan pour définir une image d'arrière-plan de panneau par programmation.



**Remarque** Si vous sélectionnez une image qui n'apparaît pas dans la liste **Arrière-plan**, LabVIEW ne l'ajoute pas à la liste **Arrière-plan** de façon permanente. Pour ajouter une image à la liste de façon permanente, vous devez enregistrer cette image dans le répertoire `labview\resource\backgrounds`.

Lorsque vous enregistrez un VI qui comporte un panneau doté d'une image en arrière-plan, LabVIEW enregistre l'image d'arrière-plan du panneau avec le VI.

### Verrouillage de la course des boutons rotatifs au minimum et au maximum

Par défaut, les boutons rotatifs et les cadrans ne peuvent pas être actionnés au-delà de leur valeur minimale ou maximale.

Pour désactiver le verrouillage, cliquez avec le bouton droit sur le bouton rotatif, sélectionnez **Propriétés** dans le menu local et désélectionnez l'option **Verrouiller au minimum et au maximum**. Verrouiller le bouton rotatif l'empêche les valeurs de passer brusquement du minimum au maximum ou vice versa. Désactiver cette option peut engendrer des sauts de valeurs inattendus.

Dans LabVIEW 8.2, lorsque vous ouvrez un VI enregistré dans LabVIEW 8.0 ou version antérieure, le verrouillage est désactivé. Pour activer le verrouillage, cochez la case **Verrouiller au minimum et au maximum**.

## Glissement de plusieurs éléments au sein de commandes arbre et de listes déroulantes

Vous pouvez sélectionner plusieurs éléments de commandes arbre et de listes déroulantes pour effectuer des opérations Glisser - Déposer. Cliquez avec le bouton droit sur la commande arbre ou la liste déroulante et sélectionnez **Mode de sélection»0 élément ou plus** ou **Mode de sélection»1 élément ou plus** dans le menu local pour activer le glissement et le dépôt de plusieurs éléments. Effectuer un glissement avec plusieurs éléments sélectionnés déplace tous les éléments sélectionnés.

## Améliorations apportées au graphe numérique

Reportez-vous au livre **Principes de base»Graphes et graphes déroulants** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur les graphes waveform numériques.

Les modifications suivantes ont été apportées aux graphes numériques dans LabVIEW 8.2.

### Définition de l'épaisseur des lignes

**Style de ligne** remplace **Position de la ligne épaisse** dans le menu local de la légende des tracés du graphe numérique. Cliquez avec le bouton droit sur la légende des tracés et sélectionnez **Style de ligne** dans le menu local pour définir l'épaisseur de la ligne. LabVIEW 8.2 intègre également une nouvelle option **Style de ligne** qui permet d'épaissir toute la ligne.

### Assombrissement des données de comparaison

Si un graphe numérique comprend des données numériques dans les états logiques de comparaison et de commande, les données de comparaison apparaissent plus foncées sur le tracé que les données de commande par défaut. Si vous ne voulez pas que les données de comparaison apparaissent foncées, cliquez avec le bouton droit sur le tracé et sélectionnez **Avancé»Assombrir les données de comparaison** dans le menu local pour désélectionner l'option. Vous pouvez aussi utiliser la propriété Assombrir les données de comparaison pour assombrir les données de comparaison par programmation.



**Remarque** Cette fonctionnalité s'applique principalement aux utilisateurs qui génèrent des signaux d'E/S numériques. Reportez-vous au livre **Principes de base»Graphes et graphes déroulants»Concepts»Personnalisation des graphes et des graphes**

déroulants sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur les données de comparaison.

## Autres améliorations apportées à la face-avant

D'autres améliorations ont été apportées à la face-avant de LabVIEW 8.2.

- Dans LabVIEW 8.0, la première fois que vous affichez un sous-titre pour un objet de la face-avant, LabVIEW déplace l'étiquette pour la mettre sur le côté. Dans LabVIEW 8.2, LabVIEW masque l'étiquette et n'affiche que le sous-titre.
- Dans la boîte de dialogue **Propriétés du graphe XY**, le menu déroulant **Afficher le plan optionnel** et les options de configuration du plan que vous sélectionnez ont été déplacées de la page **Échelles** à la page **Apparence**.
- Vous pouvez utiliser l'outil Pinceau pour changer la couleur d'arrière-plan d'une table système.
- Vous pouvez changer la couleur des en-têtes et des cellules d'une liste, d'une table, d'une commande arbre ou d'une liste multicolonne du système.
- Les options **Rétablir la valeur par défaut**, **Couper les données** et **Coller les données** ne sont pas disponibles dans le menu local d'un indicateur quand le VI est en mode exécution. Ces options de menu local ne sont disponibles que pour les commandes en cours d'exécution.
- Pour redimensionner une commande onglet pour l'adapter à son contenu, cliquez avec le bouton droit sur la commande onglet et sélectionnez **Avancé»Ajuster la taille** dans le menu local.
- Lorsque vous cliquez avec le bouton droit sur une commande de type énumération, menu déroulant, ou chaîne déroulante et que vous sélectionnez **Éditer les éléments** dans le menu local, vous devez double-cliquer dans une cellule pour éditer un élément dans la colonne **Éléments** ou **Valeurs**. Ce changement s'applique aussi quand vous effectuez un clic droit sur les étiquettes texte d'une commande glissière ou d'un bouton rotatif et sélectionnez **Éditer les éléments** dans le menu local.
- Dans LabVIEW 8.0, quand vous sélectionnez **Fichier»Appliquer les modifications** dans la fenêtre de l'Éditeur de commande, LabVIEW conserve l'étiquette, le sous-titre et la valeur de la commande d'origine. Dans LabVIEW 8.2, LabVIEW applique toutes les modifications que vous effectuez dans la fenêtre de l'Éditeur de commande mais ne conserve aucune information de la commande d'origine. Ce comportement s'applique uniquement aux commandes personnalisées et non aux définitions de type.

- LabVIEW n’inclut pas les tracés masqués lorsque la mise à l’échelle automatique est activée pour les axes d’un graphe ou d’un graphe déroulant. Si vous souhaitez inclure les tracés masqués lors de la mise à l’échelle automatique, rendez-les transparents. Cliquez avec le bouton droit sur la légende du tracé et sélectionnez **Couleur** dans le menu déroulant pour changer la couleur des tracés.
- **(Windows)** Les cases à cocher et les boutons radio dans LabVIEW ont le même comportement que ceux de Windows. Vous pouvez actionner les boutons radio et cocher les cases à cocher en utilisant la barre espace. Vous pouvez actionner les boutons des boîtes de dialogue avec la touche <Entrée> du clavier alphanumérique, avec la touche <Entrée> du pavé numérique ou avec la barre espace. Les raccourcis clavier de LabVIEW <T> et <F> n’actionnent pas les boutons des boîtes de dialogue.
- LabVIEW 8.2 inclut de nouveaux symboles permettant l’édition avancée de la commande d’horodatage. Cliquez avec le bouton droit sur la commande puis sélectionnez **Propriétés** dans le menu local pour afficher la boîte de dialogue **Propriétés d’horodatage**. Sur la page **Format et précision**, sélectionnez l’option **Mode d’édition avancé** pour afficher la liste **Codes de format de temps absolu**. LabVIEW 8.2 inclut les nouveaux codes de formats suivants.

Code de format	Valeur
<%^<>T>	Conteneur de temps universel
<%z>	Différence entre l’heure locale et le temps universel

## Améliorations apportées à l’environnement

Les améliorations suivantes ont été apportées à l’environnement de LabVIEW 8.2.

### Enregistrement automatique pour la récupération

En cas d’arrêt anormal ou de panne système, LabVIEW sauvegarde dans un emplacement temporaire tout VI modifié (.vi), VI modèle (.vit), commande (.ctl) ou modèle de commande (.ctt) ouvert au moment de la panne ou de l’arrêt. LabVIEW ne sauvegarde pas les projets (.lvproj), les bibliothèque de projet (.lvlib), les commandeX (.xctl) ou les classes LabVIEW (.lvclass).

Si LabVIEW enregistre automatiquement des fichiers avant un arrêt anormal ou une panne système, la fenêtre **Sélectionner les fichiers à récupérer** apparaît la prochaine fois que vous lancez LabVIEW. Sélectionnez le fichier que vous voulez récupérer et cliquez sur le bouton **Récupérer**. Si vous ne souhaitez récupérer aucun fichier, désélectionnez

tous les fichiers et cliquez sur le bouton **Supprimer**. Cliquez sur le bouton **Annuler** pour déplacer tous les fichiers sélectionnés dans le sous-répertoire `LVAutoSave\archives` du répertoire de données par défaut.

Sélectionnez **Outils»Options** puis **Environnement** dans la liste **Catégorie** pour activer ou désactiver l'enregistrement à des fins de récupération et pour spécifier la fréquence de la sauvegarde automatique.

## Améliorations apportées aux boîtes de dialogue

Les améliorations suivantes ont été apportées aux boîtes de dialogue de LabVIEW 8.2 :

### Améliorations et modifications apportées à la boîte de dialogue Options

Les améliorations suivantes ont été apportées à la boîte de dialogue **Options** de LabVIEW 8.2. Reportez-vous au site Web de National Instruments sur [ni.com/fr/info](http://ni.com/fr/info) et entrez l'info-code `ex6rkc` pour obtenir davantage d'informations sur ces options obsolètes.

- La page **Performances et disque** n'existe plus, ainsi que les options **Vérifier l'espace disque disponible lors du lancement** et **Exécuter en utilisant plusieurs threads**.
- Sur la page **Face-avant**, les options **Ignorer les touches de fonction par défaut du système** et **Utiliser le rafraîchissement progressif pendant le tracé** n'existent plus.
- Sur la page **Diagramme**, l'option **Guides de câblage** n'existe plus.
- Sur la page **Chemins**, l'option **Répertoire des fonctions** du menu déroulant n'existe plus.
- Sur la page **Environnement**, l'option **Menus abrégés** n'existe plus.
- Sur la page **Environnement**, la case **Activer la boîte de dialogue Astuce** n'est plus cochée par défaut.
- Sur la page **Diagramme**, la case **Affichage du nom des sous-VIs dans le diagramme** s'applique également aux variables globales. Assurez-vous que cette option est cochée si vous voulez afficher l'étiquette de la variable globale quand vous la placez sur le diagramme.
- Sur la page **Serveur Web : VIs visibles**, l'option **VI sélectionné** s'appelle désormais **VI visible**.

## Améliorations apportées aux gestionnaire de LLBs

Les améliorations suivantes ont été apportées à la fenêtre du **Gestionnaire de LLBs**.

- Un bouton **Parcourir** apparaît à droite du champ **Répertoire**. Cliquez sur ce bouton pour naviguer vers la LLB que vous voulez modifier.
- Un bouton **Supprimer** se trouve dans la barre d'outils. Sélectionnez un fichier dans la liste **Fichiers** et cliquez sur ce bouton pour supprimer le fichier sélectionné de la LLB.
- Vous pouvez classer par colonne dans la liste **Fichiers** en cliquant sur les colonnes.

## Page Liaison des données

Les améliorations suivantes ont été apportées à la page **Liaison des données** de la boîte de dialogue **Propriétés** de toutes les commandes de la face-avant de LabVIEW 8.2 :

- Les champs **Élément réseau sélectionné** et **Élément du projet sélectionné** ont été remplacés par le champ **Chemin**.
- La section **Mode** a été remplacée par le menu déroulant **Type d'accès**.
- La case **Clignoter quand l'alarme est déclenchée** n'apparaît plus.
- Le bouton **Parcourir** qui apparaît quand vous sélectionnez **Moteur de variable partagée (NI-PSP)** dans le menu déroulant **Sélectionner l'élément liaison des données** lance la boîte de dialogue **Sélectionner l'élément source** au lieu de la boîte de dialogue **Sélectionner un élément du réseau**. La boîte de dialogue **Sélectionner un élément du réseau** n'existe plus.
- Le menu déroulant **Source publiée sur réseau** apparaît dans la boîte de dialogue **Sélectionner l'élément source** au lieu d'apparaître sur la page **Liaison des données**.

## Autres améliorations apportées aux boîtes de dialogue

Les améliorations supplémentaires suivantes ont été apportées aux boîtes de dialogue de LabVIEW 8.2 :

- La fenêtre **Statistiques de VI** intègre une ligne dans la table **Statistiques de mesures** pour afficher la moyenne de toutes les valeurs d'une colonne. Vous pouvez aussi exclure les fichiers situés dans un dossier spécifique en cochant l'option **Exclure les fichiers de ce dossier des statistiques**.
- Dans la boîte de dialogue **Propriétés du VI**, la page **Sécurité** s'appelle désormais **Protection**.
- La fenêtre **VI polymorphe** comprend un bouton **Afficher la mise en garde de licence** qui apparaît si le VI polymorphe appartient à une

bibliothèque de projet sous licence où la licence est soit en mode évaluation ou non valide. Cliquez sur ce bouton pour afficher un message d'avertissement. Cliquez sur le bouton **Aide** dans le message d'avertissement pour obtenir des informations complémentaires sur l'état de la licence.

- Dans la boîte de dialogue **Éditer la chaîne de format** et la boîte de dialogue **Éditer la chaîne de balayage**, vous pouvez choisir la notation SI dans le menu local **Opération sélectionnée** comme type de conversion.
- Dans la boîte de dialogue **Exporter une image simplifiée**, l'option **Enregistrer sur le presse-papiers** a été renommée **Exporter vers le presse-papiers**.
- Dans la boîte de dialogue **Création d'un VI de driver d'instruments**, **Aucun** n'est plus une option du menu déroulant **Type de données en sortie** sur la page **Configuration de commande**.
- Dans la fenêtre **Éditeur de fichier des codes d'erreur**, l'option **Code d'erreur et description actuels** a été remplacée par les options **Code d'erreur** et **Texte d'erreur**.

## Affichage des commandes et indicateurs masqués

Vous pouvez afficher toutes les commandes et tous les indicateurs masqués sur la face-avant des commandes personnalisées et des variables globales en sélectionnant **Édition**»**Afficher les commandes et indicateurs masqués**. Cette option n'est disponible que dans le menu **Édition** d'une commande personnalisée ou d'une variable globale.

Vous pouvez aussi afficher des commandes et des indicateurs pour des VIs autres que des commandes personnalisées ou des variables globales en exécutant le VI ShowHidden Core situé dans le répertoire `labview\project\_ShowHidden`.

## Améliorations apportées à la fenêtre Liste des erreurs

Dans la liste **Éléments avec erreurs** de la fenêtre **Liste des erreurs**, les éléments brisés apparaissent avec un symbole x à côté du nom de l'élément. LabVIEW classe ces éléments en haut de la liste **Éléments avec erreurs**. Les éléments qui provoquent des erreurs dans d'autres éléments car ils sont en cours de modification ont une icône crayon qui apparaît à côté du nom de l'élément. Les éléments dont le nom n'est pas juxtaposé à une icône ont été brisés car un élément dont ils dépendent a des erreurs.

## Améliorations apportées à Rechercher et remplacer

Les opérations rechercher et remplacer de LabVIEW 8.2 et les fonctionnalités associées présentent les améliorations suivantes.

- Dans les **Options de recherche de texte**, l'option **Ignorer les clones** ignore tout clone de la face-avant ou des faces-avant de VIs réentrants lorsque vous recherchez du texte dans un VI.
- Dans la boîte de dialogue **Rechercher**, le bouton **Sélectionnez l'objet** apparaît avec le titre de l'objet que vous sélectionnez au lieu du nom de fichier de l'objet.
- La fenêtre **Résultats de la recherche** apparaît seulement si LabVIEW trouve plusieurs objets au cours de la recherche. Si LabVIEW trouve seulement un objet, il le met en évidence sur la fenêtre de la face-avant ou sur la fenêtre du diagramme.
- La boîte de dialogue **Rechercher** permet aussi de rechercher et remplacer des VIs Express et des nœuds comme des variables et la fonction Rechercher une expression régulière.

## Gestion des fenêtres ouvertes

Le menu **Fenêtre** affiche jusqu'à 10 fenêtres ouvertes. Vous pouvez gérer toutes les fenêtres ouvertes en sélectionnant **Fenêtre»Toutes les fenêtres** ou en appuyant sur <Ctrl-Maj-W> pour afficher la boîte de dialogue **Toutes les fenêtres**. Vous pouvez afficher ou fermer une fenêtre, ou enregistrer l'élément qui correspond à cette fenêtre en cliquant sur le bouton correspondant sur le côté droit de la boîte de dialogue.

Les fenêtres modifiées depuis leur dernier enregistrement comportent un astérisque après leur nom dans la colonne de **Titre**.

## Améliorations apportées aux palettes

Les améliorations suivantes ont été apportées aux palettes de LabVIEW 8.2 :

- Vous pouvez utiliser la boîte de dialogue **Organiser les favoris** pour changer l'ordre des éléments dans la catégorie de palette **Favoris**.
- La boîte de dialogue **Documentation de menu** est devenue la boîte de dialogue **Documentation sur la palette**. Le champ **Description du menu** s'appelle désormais **Description de la palette**.
- Si une palette appartient à une bibliothèque de projet, vous pouvez afficher le chemin de la bibliothèque du projet. Sélectionnez **Outils»Avancé»Éditer le jeu de palettes**, cliquez avec le bouton droit sur une palette et sélectionnez **Afficher le chemin vers le fichier palette** dans le menu local. LabVIEW affiche le chemin réel de la

palette et le chemin de la bibliothèque propriétaire si la palette appartient à une bibliothèque.

- La boîte de dialogue **Éditer le jeu de palettes de commandes et de fonctions** comprend l'option **Aperçu des modifications avant l'enregistrement**. Cochez l'option **Aperçu des modifications avant l'enregistrement** et cliquez sur le bouton **Enregistrer les modifications** pour afficher la boîte de dialogue **Aperçu des modifications de palette**.
- L'élément de menu local **Déposer le VI** a été renommé **Placer le VI**. Cet élément de menu local apparaît si vous cliquez avec le bouton droit sur un VI de la palette **Fonctions**.
- LabVIEW 8.2 comprend des palettes qui restent vides jusqu'à ce que vous installiez un logiciel supplémentaire. Par exemple, la sous-palette **Conception et simulation de contrôle** de la palette **Commandes** est vide jusqu'à ce que vous installiez l'un des produits de Conception et simulation de contrôle de LabVIEW.

## Enregistrement d'une hiérarchie dupliquée

Vous pouvez enregistrer un VI et ses sous-VIs comme hiérarchie dupliquée sans devoir créer de source de distribution. Sélectionnez **Fichier»Enregistrer sous** et sélectionnez **Dupliquer la hiérarchie à un nouvel emplacement** pour enregistrer le VI et sa hiérarchie dans un nouvel emplacement.

## Autres améliorations apportées à l'environnement

D'autres améliorations ont été apportées à la face-avant de LabVIEW 8.2 :

- LabVIEW n'affiche plus de filigrane sur les sous-VIs dans une face-avant secondaire d'un VI en mode d'évaluation ou dans les versions pour étudiants de LabVIEW. LabVIEW n'affiche plus non plus de filigrane sur la version de mise au point du déploiement de LabVIEW.
- Si vous ne possédez pas de licence valide spécifique à la version de LabVIEW que vous avez achetée, il se peut que vous ne soyez pas en mesure de modifier des VIs polymorphes.
- Vous pouvez créer une référence à une commande dans une définition de type stricte en cliquant avec le bouton droit sur la commande dans la définition de type et en sélectionnant **Créer»Référence** dans le menu local.
- Le menu **Affichage** comprend les éléments **Bibliothèque de ce VI**, **CommandeX de ce VI** ou **Classe de ce VI** qui met en surbrillance la bibliothèque, la commandeX ou la classe LabVIEW à laquelle le VI actuel appartient dans la fenêtre de l'**Explorateur de projet**. L'élément de menu change en fonction du type de bibliothèque

propriétaire du VI. Si la bibliothèque, la commandeX ou la classe ne se trouve pas dans le projet LabVIEW, LabVIEW ouvre une nouvelle fenêtre contenant seulement la bibliothèque, la commandeX ou la classe à laquelle appartient le VI actuel.

- Dans LabVIEW 8.0, l'élément de menu **Aide»Expliquer l'erreur** a été supprimé de la fenêtre **Démarrage**. Dans LabVIEW 8.2, ce menu est disponible dans la fenêtre **Démarrage**.
- **(Mac OS)** Pour afficher ou masquer la fenêtre d'**Aide contextuelle**, sélectionnez **Aide»Afficher l'aide contextuelle** ou appuyez sur les touches <Commande-Maj-H>. Le raccourci clavier <Commande-H> masque l'application LabVIEW ouverte. Reportez-vous à l'*Aide LabVIEW* pour obtenir davantage d'informations sur la personnalisation des raccourcis clavier.
- Sélectionnez **Outils»Instrumentation»Développement avancé** pour accéder aux options de développement avancé de LabVIEW en utilisant les drivers d'instruments, notamment :
  - **Afficher les spécifications de driver** affiche le manuel Instrument Driver Guidelines à partir de la base de drivers d'instruments de National Instruments sur [ni.com/idnet](http://ni.com/idnet) dans un navigateur Web.
  - **Afficher le glossaire des graphiques** affiche l'Icon Art Glossary à partir de la base de drivers d'instruments de National Instruments sur [ni.com/idnet](http://ni.com/idnet) dans un navigateur Web.
  - **Autres ressources** affiche les ressources et les outils de développement à partir de la base de drivers d'instruments de National Instruments sur [ni.com/idnet](http://ni.com/idnet) dans un navigateur Web.

## Améliorations apportées aux nœuds ; nouveautés et modifications apportées aux VIs et fonctions

Les VIs et fonctions de LabVIEW 8.2 comprennent les nouveautés et les modifications suivantes. Reportez-vous au livre **Référence des VIs et fonctions** dans l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur les VIs, fonctions et les nœuds.

# Nouveaux VIs et fonctions

LabVIEW 8.2 inclut les nouveaux VIs et fonctions suivants.

## VIs de fichiers avancés

La palette **Fonctions de fichiers avancées** comporte les nouveaux VIs suivants :

- VI Vérifier si le fichier ou le dossier existe
- VI Comparer deux chemins
- VI Générer un chemin de fichier temporaire
- VI Obtenir l'extension d'un fichier
- VI Empreinte numérique MD5 du fichier
- VI Liste récursive des fichiers

## Fonctions de Waveforms numériques

La palette **Waveform numérique** comprend les nouvelles fonctions suivantes :

- Construire une waveform
- Construire des données numériques
- Obtenir les composantes d'une waveform
- Obtenir des composantes de données numériques

## VIs Mathématiques

La palette **Mathématiques** comporte les nouveaux VIs suivants dans les systèmes de développement complet et de développement professionnel de LabVIEW.

- VI Produit de Kronecker
- VI Équations de Lyapunov

## VIs .NET

La palette **.NET** contient les nouveaux VIs suivants :

- En objet .NET
- Objet.NET en variant

## VI de mise à l'échelle

La palette **Mise à l'échelle** comporte les nouveaux VIs suivants :

- VI Convertir une mesure de RTD
- VI Convertir une mesure de jauge de contrainte

- VI Convertir une mesure de thermistance
- VI Convertir une mesure de thermocouple

## **VIs de traitement de signaux**

La palette **Traitement du signal** comporte les nouveaux VIs suivants dans les systèmes de développement complet et de développement professionnel de LabVIEW :

- VI Fenêtre de Bohman
- VI Décimer (en continu)
- VI Décimer (opération unique)
- VI Filtre RIF
- Filtre RIF avec I.C.VI
- VI Motif sinusoïdal modulé gaussien
- VI Impulsion gaussienne unique
- VI Transformée en Chirp-Z inverse
- VI Fenêtre de Bartlett-Hanning modifiée
- VI Fenêtre de Parzen
- VI Motif sinc périodique
- VI Train d'impulsions
- VI Rééchantillonnage rationnel
- VI Filtre de Savitzky-Golay
- VI Coefficients du filtre de Savitzky-Golay
- VI Motif triangulaire
- VI Suréchantillonnage
- VI Fenêtre de Welch

## **VIs et fonctions TDM en continu**

La palette **TDM en continu** comporte les nouveaux VI et fonctions suivants :

- VI TDMS - Afficheur de fichier
- Fonction TDMS - Fermer
- Fonction TDMS - Défragmenter
- Fonction TDMS - Vider
- Fonction TDMS - Obtenir des propriétés
- Fonction TDMS - Répertoire le contenu
- Fonction TDMS - Ouvrir

- Fonction TDMS - Lire
- Fonction TDMS - Définir des propriétés
- Fonction TDMS - Écrire

La palette **Stockage** comporte les nouveaux VIs suivants :

- VI Convertir TDM en TDMS
- VI Convertir TDMS en TDM

Reportez-vous à la section [Format de fichier TDMS en continu](#) de ce document pour obtenir des informations sur le format de fichier TDM en continu.

## Nœuds, fonctions et VIs modifiés

Les fonctions nœuds et VIs suivants ont été modifiés dans LabVIEW 8.2 :

### Fonctions DataSocket

Les VIs suivants ont été modifiés dans la palette **DataSocket** :

- Vous pouvez activer les notifications synchrones pour les éléments de données NI-PSP lorsque vous utilisez la fonction DataSocket Écrire. En ajoutant `?sync="true"` à la fin de l'URL `psp` quand vous activez les notifications synchrones, vous pouvez spécifier une valeur **timeout en ms** différente de zéro. La fonction attend jusqu'à ce que l'opération soit terminée ou que le timeout ait expiré. L'activation des notifications synchrones peut réduire les performances, notamment les cibles RT.
- La fonction DataSocket Lire a une sortie **état** qui renvoie les mises en garde et les erreurs d'un serveur PSP ou d'un contrôleur FieldPoint.

### VIs Mathématiques

Dans la version de base de LabVIEW, le VI  $A \times B$  comporte de nouvelles instances : Vecteur  $x \times A$  et Vecteur  $x \times A$  complexes.

Les VIs suivants ont été modifiés dans la palette **Mathématiques** des systèmes de développement complet et de développement professionnel de LabVIEW.

- Le VI Ajustement exponentiel comporte une nouvelle entrée **Pondération** qui spécifie le tableau de pondérations pour les observations ( $X$ ,  $Y$ ). Les valeurs négatives résultant fréquemment du bruit du signal ne font plus échouer l'ajustement exponentiel.
- Le VI PGCD de  $p(x)$  et  $q(x)$  comporte une nouvelle entrée **algorithme** qui spécifie l'algorithme utilisé par le VI pour calculer le plus grand commun diviseur.

- Le VI Ajustement polynomial général comporte une nouvelle entrée **Pondération** qui spécifie le tableau de pondérations pour les observations (**X**, **Y**).
- Les VIs Histogramme, Histogramme PPP, Histogramme général et Histogramme général PPP comportent une sortie **Grphe de l'histogramme** qui affiche le diagramme à barres de l'histogramme de la séquence **X** en entrée. L'axe des y correspond au compte de l'histogramme et l'axe des x aux valeurs centrales des intervalles de l'histogramme.
- Le VI PPCM de  $p(x)$  et  $q(x)$  comporte une nouvelle entrée **algorithme** qui spécifie l'algorithme utilisé par le VI pour calculer le plus petit commun multiple des polynômes.
- Le VI Développement en fractions partielles comporte une nouvelle entrée **option** qui spécifie comment le VI gère les cofacteurs de **Numérateur** et **Dénominateur**.
- Le VI Équations de Sylvester comporte une nouvelle entrée **type de la matrice** qui spécifie le type des entrées **A** et **B**, ce qui accélère le calcul de **X**.
- L'entrée **Cosinus directeurs** de l'instance Tableau du VI Rotation de coordonnées cartésiennes 3D (direction) est devenue **Matrice de rotation**.

## Fonctions numériques

Les fonctions suivantes ont été modifiées dans la palette **Numérique** :

- La fonction Quotient et Reste renvoie des réponses plus précises lorsque vous utilisez les grands diviseurs négatifs 64 bits.
- L'entrée **données quelconques** des fonctions Permuter les octets et Permuter les mots est devenue **données**. Vous pouvez câbler des chaînes, des tags, des chemins, des booléens, des nombres non entiers, des énums, des clusters d'erreur, des images, des matrices, et des données clusters et tableau de ces types et de les transmettre sans modification. Pour tout entier 16, 32 ou 64 bits câblé à l'entrée, la fonction intervertit chaque paire d'octets dans chaque mot ou dans chaque paire de mots dans chaque mot long. Vous ne pouvez pas câbler d'occurrences, de types de refnum ou de variants à l'entrée.
- Sur les plates-formes x86, la fonction Multiplier par une puissance de 2 produit la réponse exacte lorsque les entrées **x** et **n** sont des nombres à virgule flottante.

## VIs et fonctions de protocoles

Les fonctions et VIs suivants ont été modifiés dans la palette **Protocoles** :

- La fonction UDP Ouvrir et le VI UDP Multicast - Ouvrir comportent une nouvelle sortie **port** qui renvoie le numéro du port que la fonction a utilisé.
- La fonction UDP Ouvrir et le VI UDP Multicast - Ouvrir ont une nouvelle entrée **adresse réseau**, sur quelle adresse du réseau écouter.
- L'entrée **adresse** de la fonction TCP Ouvrir une connexion est passée d'optionnelle à recommandée.
- Le VI TCP Écouter comporte une entrée **résoudre l'adresse à distance (V)** qui indique s'il faut appeler la fonction IP en chaîne sur l'adresse distante. La valeur par défaut est VRAI.

## VI Programmation quadratique

Le VI Programmation quadratique comprend les modifications suivantes :

- Le VI Programmation quadratique est un VI polymorphe aux instances suivantes : Programmation quadratique - PI et Programmation quadratique - EA. L'instance Programmation quadratique - PI a la même fonctionnalité que le VI Programmation quadratique de LabVIEW 8.0. L'instance Programmation quadratique - EA recherche le minimum en utilisant l'algorithme Ensemble actif.
- Les deux instances du VI Programmation quadratique comportent une entrée **début** qui spécifie le point dans la dimension  $n$  auquel le processus d'optimisation commence.
- Les deux instances du VI Programmation quadratique comportent une entrée **temps max (s)** qui spécifie la durée maximale que LabVIEW alloue entre le début et la fin du processus d'optimisation.
- L'instance Programmation quadratique - EA du VI Programmation quadratique comporte une entrée **démarrage à chaud ?** qui indique que le démarrage à chaud de l'optimisation est permis.

## VIs de traitement de signaux

Les VIs suivants ont été modifiés dans la palette **Traitement du signal** des systèmes de développement complet et de développement professionnel de LabVIEW.

- Le VI Filtre médian comporte les nouvelles entrées **rang de gauche** et **rang de droite** qui remplacent l'entrée **rang** de LabVIEW 8.0. Ce VI applique un filtre médian de rang à la séquence **X** en entrée, où rang est **rang de droite** si **rang de droite** est supérieur à zéro, ou **rang de gauche** si **rang de droite** est inférieur à zéro.



**Remarque** Si vous ouvrez un VI créé de LabVIEW 8.0 ou version antérieure qui contient le VI Filtre médian avec une valeur câblée à l'entrée **rang**, LabVIEW utilise cette valeur pour **rang de gauche** dans LabVIEW 8.2.

- L'entrée **f** des VIs Signal sinusoïdal, Signal sinusoïdal PPP, Signal triangulaire, Signal triangulaire PPP, Signal en dents de scie, Signal en dents de scie, Signal carré, Signal carré PPP et Signal arbitraire est devenue **fréquence**.
- L'entrée **nb de points latéraux** du VI Filtre de Savitzky Golay PPP a été changée en **points de côté**.
- Les VIs Point par Point ont de nouvelles icônes.
- L'entrée **fenêtre** des VIs Fenêtre symétrique et Fenêtre temporelle mise à l'échelle comporte les nouvelles valeurs suivantes : Bartlett-Hanning modifiée, Bohman, Parzen et Welch.

## VIs de son (Linux)

Vous devez avoir le driver OSS (Open Sound System) pour utiliser les VIs de **Son**. Reportez-vous au livre **Informations importantes** » **Copyright** de l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations sur le copyright OSS applicable.



**Remarque** LabVIEW recherche les périphériques en cherchant les fichiers nommés `/dev/dsp` ou `/dev/dspX` où *X* est un entier compris entre 0 et 16. LabVIEW essaie d'ouvrir chaque périphérique pour les entrées et sorties. Si LabVIEW ne peut pas détecter la carte son, vérifiez qu'un fichier de périphérique appelé `/dev/dsp` ou `/dev/dspX` se trouve bien sur le système local et que vous avez l'autorisation de lire et d'écrire sur ce périphérique. Si vous avez déplacé ce périphérique dans un emplacement autre que l'emplacement par défaut, LabVIEW peut fonctionner avec un lien symbolique.

Les VIs de Son sous Linux intègrent les modifications suivantes :

- Les VIs supportent les sons mono et stéréo.
- Une waveform représente des données de son. Vous pouvez utiliser des éléments de type entiers non signés 8 bits, signés 16 bits ou signés 32 bits, ou de type de données double ou simple précision pour représenter les données du tableau **Y**. Chaque waveform définit une voie.
- Les données de son sont au format PCM (Pulse Code Modulated).
- Les VIs peuvent générer du son en continu.
- Les VIs permettent un affichage continu des fichiers wave.
- La fonctionnalité de vérification des erreurs des VIs a été améliorée.

## Fonctions de chaîne

Les fonctions suivantes ont été modifiées dans la palette **Chaîne** :

- L'entrée **chaîne de format** de la fonction Balayer une chaîne accepte les valeurs en notation SI (System International).
- La fonction Chaîne au format tableur en tableau fonctionne correctement quand l'entrée **type de tableau** est numérique complexe.

## Fonctions VISA

Lorsque vous cliquez avec le bouton droit sur les terminaux de certaines fonctions VISA et que vous sélectionnez **Créer»Constante**, **Créer»Commande** ou **Créer»Indicateur** dans le menu local, un objet menu déroulant apparaît. Les objets apparaissent sous forme d'un menu déroulant que vous pouvez parcourir pour effectuer des sélections. Toutes les valeurs que LabVIEW supporte apparaissent dans les menus déroulants.

Cette modification affecte les fonctions et les terminaux suivants.

Fonction	Terminal(aux)
VISA Assert Interrupt Signal	mode
VISA Assert Trigger	protocole
VISA Assert Utility Signal	signal de bus
VISA Disable Event	type d'événement
VISA Discard Events	type d'événement
VISA Enable Event	type d'événement
VISA Find Resource	mode de recherche
VISA GPIB Control ATN	mode
VISA GPIB Control REN	mode
VISA Map Trigger	source de déclenchement, déclenchement de destination
VISA Open	mode d'accès
VISA Unmap Trigger	source de déclenchement, déclenchement de destination
VISA VXI Cmd or Query	mode
VISA Wait On Event	type d'événement en entrée, type d'événement en sortie

## Appeler une fonction d'une DLL

Le nœud Appeler une fonction d'une DLL comporte des terminaux d'erreur. La boîte de dialogue **Appeler une fonction d'une DLL** comprend également les modifications suivantes :

- La boîte de dialogue comprend plusieurs pages que vous pouvez utiliser pour configurer des paramètres plus facilement.
- Sur la page **Fonction**, vous pouvez configurer le nœud pour pouvoir spécifier par programmation le chemin de la bibliothèque en cochant la case de **Spécifier le chemin sur le diagramme**.
- Sur la page **Paramètres**, vous pouvez utiliser le champ **Taille minimale** lorsque vous configurez des paramètres chaîne ou tableau pour allouer correctement de la mémoire.
- Sur la page **Callbacks**, vous pouvez spécifier des callbacks définis par utilisateur.

## Autres modifications apportées aux VIs, fonctions et nœuds

Les autres modifications suivantes ont été apportées aux fonctions, VIs et nœuds de LabVIEW 8.2 :

- Les options **Opération mathématique** de la boîte de dialogue de configuration du VI Express Opérations dans le domaine temporel sont passées de **Différentielle** à **Dérivée (dX/dt)**, de **Différence** à **Différence (dX)**, de **Intégrale** à **Intégrale (Somme[Xdt])** et de **Somme** à **Somme (Somme[X])**.
- **(Windows)** L'entrée **refnum en entrée** de l'instance Infos sur le fichier son (refnum) du VI Infos sur le fichier son a été renommée **refnum de fichier son**.
- **(Mac OS, Linux)** Le VI Set Report Footer Text génère des pieds de page de taille légèrement différente que dans les versions antérieures de LabVIEW. Le VI Définir le texte de pied de page du rapport comporte aussi une entrée **taille du pied de page HTML**.
- Le VI Synchroniser les démarrages des structures cadencées comporte une entrée **effacer** qui supprime toutes les structures cadencées et tout le groupe avant d'ajouter les structures cadencées que vous spécifiez au groupe. Utilisez cette entrée pour supprimer les structures cadencées qui ne correspondent pas à une boucle cadencée. Ce VI comporte aussi une nouvelle sortie, **noms des structures cadencées en sortie**, qui renvoie le nom de toutes les structures cadencées dans le groupe après que LabVIEW ait ajouté les noms au groupe de synchronisation.
- **(Windows)** Dans LabVIEW 8.0, la fonction Lister le contenu du dossier ajoutait `. *` à la fin de l'entrée **filtre** si **filtre** ne comportait pas d'extension. Dans LabVIEW 8.2, la fonction ne modifie pas l'entrée

**filtre**, mais ajoute . au nom de fichier si le nom de fichier ne contient pas d'extension mais que **filtre** en a une.

- Une fois que vous avez placé la fonction Référence de VI statique sur le diagramme, double-cliquez dessus pour afficher une boîte de dialogue de fichier dans laquelle vous pouvez sélectionner un VI. Dans LabVIEW 8.0, double-cliquer sur la fonction affiche une boîte de dialogue indiquant qu'un fichier manque.
- L'entrée **chaîne de format** des fonctions Formater dans un fichier, Formater en chaîne, Balayer un fichier et Balayer une chaîne n'est plus une entrée requise.
- Ouvrir un VI contenant un nœud de variable partagée dans un projet où le nœud de variable partagée ne peut pas trouver la variable partagée à laquelle il est lié dans la fenêtre de l'**Explorateur de projet** fait que le nœud de la variable partagée ne fonctionne pas. Toute commande de la face-avant associée à la variable partagée manquante cesse aussi de fonctionner. Ce comportement est spécifique à Windows et se produit seulement lorsque vous ouvrez le nœud de VI dans un projet. Si vous ouvrez le VI dans l'instance d'application principale, vous ne recevez pas de notification indiquant que des variables partagées sont manquantes. Des versions antérieures de LabVIEW n'indiquaient pas que le nœud de la variable partagée ne pouvait pas trouver les variables auxquelles il était associé dans la fenêtre de l'Explorateur de projet.
- Dans LabVIEW 8.0.1, la sortie **Raison du réveil** de la Boucle cadencée est un menu déroulant. Dans LabVIEW 8.2, la sortie est de type énumération, ce qui est cohérent avec le comportement de LabVIEW 8.0.
- Les noms des sorties dupliquées sur les fonctions et VIs suivants sont passés de **dup [sortie]** à **[sortie] en sortie**. Ce changement n'affecte pas la fonctionnalité de ces VIs et fonctions.
  - Nœud d'appel par référence
  - Exécuter l'expression de requête
  - Infos sur le fichier/répertoire
  - Obtenir les infos d'objet
  - Obtenir une propriété
  - Nœud de méthode
  - Lister le contenu du dossier
  - Nœud de propriété
  - Rechercher un symbole
  - Refnum en ID
  - Définir une propriété

## Nouveaux événements, propriétés et méthodes

LabVIEW 8.2 comporte de nouveaux événements, propriétés, méthodes et classes de VI Serveur. Reportez-vous au livre **Fonctionnalités et changements dans LabVIEW 8.2** » **Nouveaux événements, propriétés, méthodes et classes du VI Serveur** de l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir une liste des nouveaux événements, méthodes, propriétés et classes.

LabVIEW 8.2 comprend aussi les nouvelles propriétés VISA suivantes :  
PXI/PCI Settings:Is PCI Express, PXI/PCI Settings:Maximum Link Width, PXI/PCI Settings:Actual Link Width, PXI/PCI Settings:Slot Link Width, PXI/PCI Settings:D-Star Bus Number et PXI/PCI Settings:D-Star Set.

## Améliorations apportées à MathScript LabVIEW

Reportez-vous au livre **Principes de base** » **Formules et équations** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur MathScript LabVIEW.

Les améliorations et les modifications suivantes ont été apportées à MathScript dans LabVIEW 8.2 :

### Nouvelles fonctions MathScript

LabVIEW 8.2 inclut les nouvelles fonctions MathScript suivantes. Vous pouvez utiliser ces fonctions dans la **fenêtre MathScript LabVIEW** ou dans le nœud MathScript.

- classe plots : area, bar, bar3, bar3h, barh, contour, contour3, contourf, errorbar, ezcontour, ezcontourf, ezmesh, ezmeshc, ezplot, ezplot3, ezpolar, ezsurf, ezsurf, feather, fill, fplot, gplot, meshc, pie, plotmatrix, plotyy, polar, quiver, scatter, scatter3, shg, stem3, strips, surf, treeplot et waterfall.
- classe dsp : ac2poly, ac2rc, arburg, arcov, armcov, aryule, barthannwin, bartlett, besslap, besself, bilinear, bitrevorder, blackman, blackmanharris, bohmanwin, buttap, cceps, cheblap, cheb2ap, chebwin, chirp, conv2, convmtx, corrmtx, czt, dct, dftmtx, digitrevorder, diric, downsample, dst, ellipap, eqtflength, filternorm, filtic, firgauss, firrcos, flattopwin, freqspace, gauspuls, gauswin, gmonopuls, goertzel, hann, iceps, iczt, idct, idst, impinvar, intfilt, invfreqs, invfreqz, is2rc, kaiserord, lar2rc, latc2tf, levinson, lp2bp, lp2bs, lp2hp, lp2lp, lpc, lsf2poly, maxflat, medfilt1, nuttallwin, parzenwin, phasedelay, phasez, poly2ac, poly2lsf, poly2rc, polyscale, polystab, prony, pulstran, rc2ac, rc2is, rc2lar,

rc2poly, rceps, rectpuls, rectwin, resample, residuez, rlevinson, schurrc, seqperiod, sgolay, sgolayfilt, sinc, sos2ss, sos2tf, sos2zp, sosfilt, spline, ss2sos, ss2tf, ss2zp, stepz, stmcb, tf2latc, tf2sos, tf2ss, tf2zp, tf2zpk, triang, tripuls, tukeywin, udecode, uencode, upfirdn, upsample, vco, xcorr, xcorr2, xcov, zerophase, zp2sos, zp2ss et zp2tf.

- classe support : csvread, csvwrite, dlmread, dlmwrite, labviewroot, type, uiload et what.
- classe string : eval.
- classe libraries : loadlibrary, calllib, unloadlibrary, libisloaded et libfunctionsview. Vous pouvez utiliser ces fonctions pour appeler des bibliothèques partagées à partir de la **fenêtre MathScript LabVIEW** ou du nœud MathScript. Reportez-vous au projet MathScript Shared Libraries.lvproj dans le répertoire labview\examples\MathScript\MathScript Shared Libraries pour obtenir des exemples d'appels de bibliothèques partagées à partir de MathScript.

## Diverses améliorations et modifications apportées à MathScript

Les modifications supplémentaires suivantes ont été apportées à MathScript dans LabVIEW 8.2 :

- Les performances générales de la **fenêtre MathScript LabVIEW** ont été améliorées. Le temps de compilation a été amélioré dans le nœud MathScript.
- Le moteur d'exécution LabVIEW supporte MathScript. Vous pouvez incorporer des nœuds MathScript dans les applications autonomes et les bibliothèques partagées que vous construisez avec l'Application Builder. Le moteur d'exécution LabVIEW ne supporte pas certaines fonctions MathScript. Si un script contient ces fonctions non supportées, vous devrez peut-être le modifier avant de construire une application ou une bibliothèque partagée. Reportez-vous à la rubrique *Fonctions MathScript non supportées par le moteur d'exécution LabVIEW* de l'*Aide LabVIEW* pour obtenir la liste complète des fonctions MathScript qui ne sont pas supportées.
- Lorsque vous appelez la commande MathScript help, LabVIEW affiche l'aide dans une fenêtre d'aide HTML. Pour afficher l'aide dans la **Fenêtre de résultat** de la **fenêtre MathScript LabVIEW**, sélectionnez **Fichier»Préférences MathScript** et décochez l'option **Afficher l'aide HTML ?**. Si vous définissez une fonction, LabVIEW continue à afficher l'aide pour la fonction définie dans la **Fenêtre de résultat**.

- Le nœud MathScript ne se comporte pas comme les autres nœuds de script lorsque vous câblez un type de données non supporté à un terminal en entrée. Sur un nœud MathScript, LabVIEW convertit le type de données en type supporté ou affiche un fil de liaison brisé. Si LabVIEW convertit le type de données, un point de coercition apparaît sur le terminal où la conversion a lieu. Après avoir câblé une entrée à un nœud MathScript, cliquez avec le bouton droit sur le terminal de l'entrée et sélectionnez **Afficher le type de données** dans le menu local pour voir le type de données de l'entrée.
- MathScript supporte tous les caractères non Unicode dans les chaînes de texte, mais pas dans les noms de variables. Vous pouvez utiliser uniquement les caractères ASCII dans les noms de variables. Par exemple, vous pouvez utiliser á dans une chaîne de texte, mais vous ne pouvez pas appeler le script suivant à partir de la **fenêtre MathScript LabVIEW** ou du nœud MathScript :

```
á=rand(50, 1)
plot(á)
```

Vous pouvez enregistrer des données dans des chemins qui contiennent des caractères non Unicode. En outre, si vous installez LabVIEW dans un chemin de répertoire qui contient des caractères non Unicode, MathScript fonctionne correctement.

- LabVIEW utilise un raccourci d'évaluation pour évaluer les expressions logiques composées dans MathScript. Par exemple, si vous exécutez la commande `if 0 == 0 || foo(a) == 2`, LabVIEW n'exécute pas `foo(a)` car la première partie de l'expression est déjà vraie. De la même façon, si vous exécutez la commande `if 0 ~= 0 && foo(a) == 2`, LabVIEW n'exécute pas `foo(a)` car la première partie de l'expression est déjà fausse.

Dans la mesure où LabVIEW 8.0 évalue toutes les parties des expressions logiques composées dans MathScript indépendamment du fait que ces expressions soient vraies ou fausses, il se peut que les scripts de LabVIEW 8.0 qui contiennent des expressions logiques composées ne s'exécutent pas comme prévu dans LabVIEW 8.2. Par exemple, dans la mesure où LabVIEW 8.0 exécute `foo(a)` dans la commande `if 0 == 0 || foo(a) == 2`, vous pouvez utiliser le résultat de `foo(a)` pour définir une variable dans un script. Ce même script s'exécute différemment dans LabVIEW 8.2. Dans la mesure où LabVIEW 8.2 n'exécute pas `foo(a)` dans la commande `if 0 == 0 || foo(a) == 2`, vous ne pouvez pas utiliser le résultat de `foo(a)` dans un script. Si vous ne voulez pas que LabVIEW utilise un raccourci d'évaluation, supprimez les expressions logiques composées de votre code existant.

- MathScript supporte les fonctions `nargin` et `nargout` des fonctions définies par l'utilisateur.

- MathScript supporte le mot-clé `return`. MathScript supporte aussi le mot-clé `end` dans une opération d'indexation de matrice.
- Les fonctions `prod` et `sum` comportent désormais une entrée **b** qui vous permet de spécifier la dimension le long de laquelle il faut calculer le produit ou la somme.

## Commande image 3D

Reportez-vous au livre **Principes de base»VIs Graphisme et son** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur la commande image 3D.

Les VIs, les propriétés et les méthodes de la palette **Contrôle d'images 3D** convertissent un groupe d'objets 3D en scène 3D que vous pouvez afficher et manipuler. Vous pouvez générer plusieurs objets 3D et spécifier leur taille, leur forme, leur mouvement, leur apparence et leur relation par rapport aux autres objets de la scène.

Utilisez les VIs, propriétés et méthodes suivants pour définir l'apparence d'un objet 3D :

- Utilisez les VIs de la palette **Objet** pour créer ou rechercher des objets 3D. Vous pouvez aussi utiliser les propriétés `ObjetScène` et les méthodes `ObjetScène` pour placer des objets 3D dans la scène et leur attribuer des caractéristiques par programmation.
- Utilisez les VIs de la palette **Chargement de fichier** pour ajouter des fichiers de modèle ou de scène existants à la scène 3D.
- Utilisez les VIs de la palette **Géométries** avec les propriétés `GéométrieScène` et les méthodes `GéométrieScène` pour spécifier la forme géométrique d'un objet 3D.
- Utilisez les VIs de la palette **Transformations** pour placer les objets dans une scène 3D.

Utilisez les VIs de la palette **Aides** pour effectuer les opérations courantes sur les scènes 3D. Vous pouvez configurer une fenêtre différente pour la scène, créer de nouveaux plans de clipping dans la scène, ajouter des sources de lumière et appliquer des textures aux objets 3D et convertir les valeurs de couleurs LabVIEW pour qu'elles apparaissent dans la commande image 3D.

Vous pouvez aussi utiliser les propriétés et méthodes suivantes pour configurer une scène 3D par programmation :

- Utilisez les propriétés `FenêtreScène` pour faire le rendu de la scène dans une autre fenêtre, configurer la fenêtre et définir l'interaction du contrôleur de caméra dans la scène.

- Utilisez les propriétés `PlanClippingScène` pour spécifier les plans de la scène dans lesquels un objet apparaît ou est masqué.
- Utilisez les propriétés `LumièreScène` pour configurer une source de lumière pour la scène.
- Utilisez les propriétés `TextureScène` et les méthodes `TextureScène` pour appliquer des textures à un objet 3D.

Reportez-vous au VI `solarsystem` dans le répertoire `labview\examples\picture\3D Picture Control` pour consulter un exemple de création de scène 3D avec la commande `image 3D`.

## Programmation orientée objet LabVIEW

Reportez-vous au livre **Principes de base»Programmation orientée objet LabVIEW** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur la programmation orientée objet dans LabVIEW.

La programmation orientée objet LabVIEW utilise des concepts d'autres langages de programmation orientés objet comme C++ et Java tels que la structure en classes, l'encapsulation et l'héritage. Vous pouvez utiliser ces concepts pour créer du code plus simple à maintenir et à modifier sans affecter d'autres sections de code au sein de l'application. Vous pouvez utiliser la programmation orientée objet LabVIEW pour créer des types de données définis par l'utilisateur.

### Création de classes LabVIEW

Pour créer des types de données définis par l'utilisateur dans LabVIEW, vous devez créer des classes LabVIEW. Les classes LabVIEW définissent les données associées à un objet ainsi que les méthodes qui définissent les actions à effectuer sur les données.

Dans LabVIEW, les données d'une classe sont privées ; autrement dit, seuls les VIs membres de la classe y ont accès. Vous définissez les données de la classe dans la commande de données privées. Lorsque vous créez et que vous enregistrez une classe LabVIEW, LabVIEW crée un fichier de bibliothèque de classe (`.lvclass`) qui définit un nouveau type de données. Le fichier de bibliothèque de classe enregistre la commande de données privées et les informations sur les VIs membres que vous créez, comme la liste des VIs et les différentes propriétés des VIs. La bibliothèque de classe est similaire à la bibliothèque de projet (`.lvlib`). Toutefois, la bibliothèque de classe définit un nouveau type de données.

Vous pouvez créer une classe en utilisant une des méthodes suivantes :

- Effectuez un clic droit sur **Poste de travail** dans la fenêtre de l'**Explorateur de projet** et sélectionnez **Nouveau»Classe** dans le menu local.
- Sélectionnez **Fichier»Nouveau** pour afficher la boîte de dialogue **Nouveau**, puis sélectionnez **Autres fichiers»Classe** dans la liste de **Créer un nouvel objet**.

## Définition de commande de données privées

LabVIEW crée automatiquement une commande de données privées pour la classe lorsque vous créez une classe LabVIEW.

Vous utilisez la fenêtre de l'Éditeur de commande pour personnaliser la commande de données privées d'une classe. LabVIEW affiche la fenêtre de l'Éditeur de commande lorsque vous double-cliquez sur la commande de données privées de la classe dans la fenêtre de l'**Explorateur de projet**. Vous pouvez déposer des commandes et des indicateurs dans le **cluster de données privées de la classe** pour définir le type de données privées d'une classe LabVIEW. Les valeurs par défaut que vous définissez pour les commandes dans le **cluster de données privées de la classe** sont les valeurs par défaut de la classe.

## Création de VIs membres

Les VIs membres mettent en œuvre les méthodes que vous créez pour la classe LabVIEW. Vous créez des VIs membres pour effectuer des opérations sur les données privées de la classe. Les VIs membres sont membres de la classe LabVIEW dans laquelle ils sont créés ; ils apparaissent dans la fenêtre de l'**Explorateur de projet** sous la commande de données privées de la classe. Vous pouvez définir la plupart des méthodes en utilisant un seul VI membre dans une classe, mais vous pouvez définir certaines méthodes en créant plusieurs VIs membres dans la hiérarchie de classes.

Vous pouvez créer un VI membre à partir d'un VI modèle qui inclut la gestion des erreurs et des objets de classe, d'un VI vide ou d'un VI membre ancêtre. Cliquez avec le bouton droit sur la classe et sélectionnez l'un des éléments suivants dans le menu local :

- **Nouveau»VI** — Ouvre un VI membre vide.
- **Nouveau»VI dynamique** — LabVIEW dote le VI membre de clusters d'**entrée d'erreur** et de **sortie d'erreur**, d'une structure Condition pour la gestion des erreurs, de la classe LabVIEW en entrée et de la classe LabVIEW en sortie.
- **Nouveau»VI de redéfinition** — Crée un VI membre qui redéfinit un VI membre ancêtre.

## Distribution d'une classe LabVIEW à d'autres développeurs et utilisateurs

Vous pouvez distribuer la classe LabVIEW que vous développez à d'autres développeurs de classes LabVIEW et à des utilisateurs de classes LabVIEW. Il existe plusieurs façons de distribuer une classe ; vous devez donc choisir celle qui répond le mieux à vos besoins. Vous pouvez utiliser l'Application Builder pour créer un fichier zip afin de distribuer une ou plusieurs classes. Vous pouvez aussi verrouiller la classe LabVIEW avant de la distribuer pour que les utilisateurs de classes LabVIEW n'aient pas accès aux données privées et aux VIs membres. En verrouillant la classe, vous évitez que les utilisateurs n'introduisent des erreurs dans l'application.

## Améliorations apportées aux projets LabVIEW

Les projets de LabVIEW 8.2 et leurs fonctionnalités associées présentent les améliorations suivantes :

### Améliorations apportées à l'Application Builder

Les améliorations suivantes ont été apportées à l'Application Builder de LabVIEW 8.2 dans le système de développement professionnel de LabVIEW :

- **Incrémentation automatique de la version du produit** — Si vous construisez un installateur plusieurs fois, LabVIEW peut incrémenter automatiquement le numéro de version du produit pour chaque nouvelle version de l'installateur. L'option **Incrémenter automatiquement la version du produit** apparaît sur la page **Informations sur le produit** de la boîte de dialogue **Propriétés de l'installateur**.
- **Spécification de la taille du média pour enregistrer l'installateur** — Vous pouvez personnaliser la taille du média lorsque vous enregistrez des composants de l'installateur. L'option **Activer la sauvegarde sur plusieurs disques** apparaît sur la page **Avancé** de la boîte de dialogue **Propriétés de l'installateur**. Le menu déroulant des médias comprend une option **Personnaliser**. Utilisez cette option pour entrer une valeur de taille de média arbitraire dans le champ **Taille du média (Mo)**.
- **Exiger Windows 2000 ou version ultérieure** — Vous pouvez exiger que les utilisateurs aient Microsoft Windows 2000 Service Pack 3 ou version ultérieure installée pour exécuter l'installateur. L'option **Windows 2000 ou version ultérieure** apparaît dans la section **Configuration système requise** de la page **Avancé** de la boîte de dialogue **Propriétés de l'installateur**.

- **Mise en cache des composants de l'installateur** — Si vous construisez un installateur plusieurs fois et que cet installateur contient des composants ou des installateurs supplémentaires, mise en cache élimine la nécessité de spécifier un emplacement pour les composants supplémentaires chaque fois que vous construisez l'installateur. La première fois que vous construisez l'installateur, la boîte de dialogue **Rechercher la distribution** vous demande de localiser la distribution qui contient les composants supplémentaires. Cochez l'option **Mettre les composants de cette distribution en cache** pour copier les fichiers de la distribution sont copiés dans un emplacement permanent sur le système local. La prochaine fois que vous construisez un installateur qui comprend ces composants, l'Application Builder copie automatiquement les composants du système local au lieu de vous demander un CD de distribution.
- **VI dynamiques et fichiers support** — Si un VI spécifié comme étant dynamique sur la page **Paramètres des fichiers source (Application)** n'est pas destiné à l'application construite, l'Application Builder déplace toutes les dépendances du VI dynamique vers le nouvel emplacement de ce VI plutôt que de les laisser dans l'application construite. Si plusieurs VIs dynamiques ou de haut niveau appellent un VI et que essaient de déplacer ce VI dans deux emplacements différents, l'Application Builder met le VI et tous ses sous-VIs dans l'application construite. Pour qu'un VI spécifié comme **Inclure uniquement si référencé** sur la page **Paramètres des fichiers source** puisse être déplacé vers un nouvel emplacement, vous devez spécifier le VI comme étant un VI dynamique.
- **Utilisation de fichiers alias de projet avec des bibliothèques partagées** — La page **Avancé** de la boîte de dialogue **Propriétés de la bibliothèque partagée** inclut l'option **Utiliser le fichier alias de projet par défaut** qui associe le fichier alias de projet à la bibliothèque partagée. Si vous décochez cette case, spécifiez un fichier alias à utiliser dans le champ **Fichier alias dans le projet**.
- **Inclure des fichiers d'en-tête LabVIEW supplémentaires avec des bibliothèques partagées** — La page **Avancé** de la boîte de dialogue **Propriétés de la bibliothèque partagée** inclut l'option **Inclure les autres fichiers d'en-tête LabVIEW**, laquelle copie tout fichier d'en-tête LabVIEW supplémentaire que référence le fichier d'en-tête généré au cours du processus de construction. Vous pouvez inclure des fichiers d'en-tête supplémentaires pour utiliser une bibliothèque partagée construite avec LabVIEW en C ou dans tout autre langage qui nécessite ces fichiers d'en-tête.
- Le bouton **Nouvelle destination** de la page **Paramètres de distribution** de la boîte de dialogue **Distribution du code source** a été renommé **Ajouter**. En outre, toutes les options liées à l'exclusion de

fichiers d'une source de distribution ont été déplacées vers la page **Autres exclusions**.

- Le bouton **Nouvelle destination** de la page **Destinations** des boîtes de dialogue **Propriétés de l'application** et **Propriétés de la bibliothèque partagée** a été renommé **Ajouter**.
- Vous pouvez construire une application ou une distribution de source sans enregistrer les VIs au préalable.
- Si vous n'ajoutez pas l'extension `.zip` aux fichiers zip que vous créez dans l'Application Builder, LabVIEW ajoute cette extension automatiquement.

## Nouvelles pages de boîtes de dialogue

Les nouvelles pages suivantes figurent dans la boîte de dialogue **Application Builder** dans LabVIEW 8.2 :

- boîte de dialogue **Propriétés de l'installateur**
  - **Informations sur les boîtes de dialogue** — Utilisez cette page pour concevoir l'interface utilisateur de l'installateur. Vous pouvez définir la langue pour le texte, afficher des fichiers licence et readme personnalisés et définir des messages et des titres de bienvenue. Cette page remplace la section **Informations sur les boîtes de dialogue** de la page **Informations sur le produit** dans LabVIEW 8.0.



**Remarque** L'option **Inclure un contrat de licence personnalisé** de la page **Informations sur les boîtes de dialogue** remplace l'option **Fichier licence** qui apparaissait sur la page **Informations sur le produit**.

- **Configuration du matériel** — Utilisez cette page pour spécifier la source des informations de configuration du matériel à inclure dans l'installateur. La nouvelle section **Mode d'importation** intègre davantage d'options permettant d'importer des fichiers de configuration du matériel à partir de Measurement & Automation Explorer. Cette page remplace la section **Configuration du matériel** de la page **Avancé** dans LabVIEW 8.0.
- boîtes de dialogue **Propriétés de l'application**, **Propriétés de la bibliothèque partagée** et **Propriétés de la distribution du code source**
  - **Autres exclusions** — Utilisez cette page pour configurer les paramètres de suppression et de déconnexion de définitions de type, d'instances de VI inutilisées et de membres non référencés des bibliothèques de projet. Utilisez ces paramètres pour réduire la taille de l'application.

## Duplication et réorganisation de spécifications de construction

Vous pouvez dupliquer des spécifications de construction dans la fenêtre de l'**Explorateur de projet**. Cliquez avec le bouton droit sur la spécification de construction à dupliquer et sélectionnez **Dupliquer** dans le menu local pour créer une copie de la spécification de construction sous **Spécifications de construction**.

Vous pouvez aussi faire glisser et déposer les éléments de la spécification de construction pour réorganiser l'ordre de construction dans la même **Spécification de construction**.

## Activation de l'Application Builder (Windows)

Si vous avez une version activée de la version de base LabVIEW ou du système de développement complet, vous pouvez sélectionner **Aide»Activer l'Application Builder** afin d'activer l'Application Builder. Les licences prendront effet après le redémarrage de LabVIEW.

## Création de bibliothèques de projet et ajout de variables partagées par programmation

Utilisez le VI CreateOrAddLibrary situé dans le répertoire `labview\vi.lib\Utility\Variable` pour ajouter par programmation une bibliothèque dans un projet ou un élément parent tel qu'un dossier, une cible ou une autre bibliothèque.

Vous pouvez aussi utiliser le VI AddSharedVariableToLibrary situé dans le répertoire `labview\vi.lib\Utility\Variable` pour ajouter des variables partagées à une bibliothèque par programmation.

## Enregistrement de projets LabVIEW et de bibliothèques de projet pour une version précédente

Vous pouvez enregistrer des projets LabVIEW et des bibliothèques de projet lisibles par LabVIEW 8.0. Pour enregistrer un projet LabVIEW pour la version précédente de LabVIEW, sélectionnez **Fichier»Enregistrer pour la version précédente** dans la fenêtre de l'**Explorateur de projet**. Pour enregistrer une bibliothèque de projet pour une version précédente, cliquez avec le bouton droit sur le fichier de la bibliothèque dans la fenêtre de l'**Explorateur de projet** et sélectionnez **Enregistrer pour la version précédente** dans le menu local, ou ouvrez la bibliothèque de projet et sélectionnez **Fichier»Enregistrer pour la version précédente**.

## Sélection d'une instance d'application

LabVIEW crée une instance d'application pour chaque cible d'un projet LabVIEW. Lorsque vous ouvrez un VI à partir de la fenêtre de l'**Explorateur de projet**, le VI s'ouvre dans l'instance de l'application de la cible. LabVIEW crée aussi une instance principale d'application, laquelle contient les VIs ouverts qui ne font pas partie d'un projet et les VIs que vous n'avez pas ouverts à partir d'un projet. Vous pouvez ouvrir des VIs dans une instance d'application spécifique en utilisant le menu local de sélection de l'instance d'application. Cliquez avec le bouton droit sur le nom de l'instance en cours dans l'angle inférieur gauche de la fenêtre de la face-avant ou du diagramme pour afficher le menu local et sélectionner une instance d'application. Si vous sélectionnez une nouvelle instance d'application, le VI est rouvert dans l'instance d'application sélectionnée. Le VI reste aussi ouvert dans l'instance d'application d'origine.

## Améliorations apportées aux variables partagées

Les améliorations suivantes ont été apportées aux variables partagées dans LabVIEW 8.2 :

- LabVIEW utilise les propriétés Réseau : taille du buffer, Réseau : taille d'élément et Réseau : points par waveform qui conviennent pour calculer la taille du buffer réseau pour une variable partagée publiée sur réseau.
  - Pour les variables partagées de type scalaire publiées sur réseau, LabVIEW utilise la propriété Réseau : taille du buffer qui indique le nombre de valeurs à buffériser.
  - Pour les variables partagées de type tableau publiées sur réseau, LabVIEW utilise les propriétés Réseau : taille du buffer et Réseau : taille d'élément.
  - Pour les variables partagées de type waveform publiées sur réseau, LabVIEW utilise les propriétés Réseau : taille du buffer et Réseau : points par waveform.
  - Pour les variables partagées tableau de waveforms publiées sur réseau, LabVIEW utilise les propriétés Réseau : taille du buffer et Réseau : points par waveform.
- La variable en entrée du nœud Variable partagée est requise quand le nœud est configuré pour écrire des données.
- La page **Variable** de la boîte de dialogue **Propriétés de la variable partagée** contient les modifications suivantes :
  - L'élément **Personnalisé** du menu déroulant **Type de données** a été renommé **De la commande personnalisée**.
  - Le menu déroulant **Type de données** comprend un nouvel élément **Variante**.

- N’affiche plus la taille de l’élément pour le buffer réseau.
- Si vous définissez **Type d’accès à Lecture seule** ou **Écriture seule**, vous pouvez créer des variables partagées qui sont configurées uniquement pour lire ou écrire des données. Lorsque vous cliquez avec le bouton droit sur la variable partagée qui est liée à une source qui est en lecture seule ou en écriture seule, LabVIEW désactive les options **Changer en écriture** et **Changer en lecture** dans le menu local.
- Dans LabVIEW 8.0, vous devez cliquer avec le bouton droit sur le nœud Variable partagée et sélectionner **Afficher l’horodatage** dans le menu local pour obtenir des informations relatives à l’horodatage sur la variable partagée à un processus. Dans LabVIEW 8.2, vous devez d’abord cliquer avec le bouton droit sur la variable partagée dans un projet, sélectionner **Propriétés** dans le menu local et cocher l’option **Activer l’horodatage** sur la page **Variable** pour obtenir des informations relatives à l’horodatage sur la variable partagée à un processus. Pour afficher les informations d’horodatage et ajouter une sortie **horodatage** au nœud Variable partagée, cliquez avec le bouton droit sur le nœud Variable partagée et sélectionnez **Afficher l’horodatage** dans le menu local. Si vous chargez dans LabVIEW 8.2 une variable partagée à processus unique créée dans LabVIEW 8.0, l’horodatage est activé par défaut.
- Dans LabVIEW 8.0, LabVIEW grise les éléments que vous ne pouvez pas sélectionner dans la liste de la boîte de dialogue **Sélectionner la variable**. Dans LabVIEW 8.2, LabVIEW grise le bouton **OK** si vous sélectionnez un élément non valide dans la liste.

## Autres améliorations apportées aux projets LabVIEW

Les autres améliorations suivantes ont été apportées aux projets dans LabVIEW 8.2 :

- Dans LabVIEW 8.0, si vous créez une définition de type dans une bibliothèque de projet et que vous définissez l’accès de la définition de type à privé, vous pouvez continuer à faire référence à la définition de type privée dans un VI en dehors de la bibliothèque de projet. Dans LabVIEW 8.2, vous ne pouvez pas faire référence à des définitions de type privées hors de la bibliothèque.
- Cliquez avec le bouton droit sur une commandeX et sélectionnez **Nouveau»VI** dans le menu local pour créer un VI dans une commandeX.
- Dans LabVIEW 8.0, vous devez enregistrer un VI de capacité, de propriété ou de méthode de commandeX après avoir défini un point d’arrêt lors de la mise au point. Dans LabVIEW 8.2, vous n’avez plus

à enregistrer les VIs de capacité, de propriété ou de méthode après avoir défini des points d'arrêt.

- Lorsque vous entrez un mot de passe pour une bibliothèque dans un projet LabVIEW qui ne possède pas de licence valide spécifique à une version de LabVIEW que vous avez achetée, vous ne pouvez pas effectuer d'opérations de Glisser - Déposer dans ou à partir de la bibliothèque.
- Si une bibliothèque de projet est protégée par un mot de passe et si ce mot de passe ne figure pas dans la mémoire cache de LabVIEW, vous pouvez effectuer un clic droit sur la bibliothèque de projet et sélectionner **Entrer le mot de passe** dans le menu local pour déverrouiller la bibliothèque du projet.

## Contrôle à distance des VIs à partir de plusieurs clients

Plusieurs clients peuvent contrôler une application ou un VI à distance en même temps. Pour permettre le contrôle simultané d'un VI, le VI doit être réentrant. Pour faire en sorte qu'un VI soit réentrant, sélectionnez **Fichier» Propriétés du VI**, puis **Exécution** dans la liste **Catégorie**, et cochez ensuite l'option **Exécution réentrante**. LabVIEW ouvre un clone du VI réentrant pour chaque demande client de face-avant distante. Vous pouvez utiliser la propriété Serveur Web : liste d'accès aux VIs pour limiter par programmation l'accès aux clones déjà en mémoire pour les connexions à des faces-avant distantes.

## Importation de fonctions à partir d'un fichier de bibliothèque partagée

Vous pouvez générer et mettre à jour des VIs pour les fonctions exportées dans un fichier `.dll` Windows, un fichier `.so` Linux ou un fichier `.framework` Mac OS. Sélectionnez **Outils»Importer»Bibliothèque partagée** puis suivez les instructions pour créer des VIs Wrapper pour les fichiers de la bibliothèque partagée. Vous devez fournir le nom d'un fichier de bibliothèque partagée et un fichier d'en-tête.

Reportez-vous à la rubrique **Principes de base»Appel de code provenant de langages de programmation textuels»Procédures»Importation de fonctions à partir d'un fichier de bibliothèque partagée** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur la création de VIs wrappers pour les fichiers de bibliothèques partagées.

## Modèles de drivers d'instruments

LabVIEW 8.2 comprend les nouveaux modèles suivants pour l'assistant Créer un nouveau projet de driver d'instruments :

- **Polyvalent (basé sur des registres)** — Utilisez ce modèle pour les instruments basés registre pour lesquels il n'y a pas de modèle classe spécifique. Des instruments basés registre courants sont le VXI et PXI.
- **Analyseur de spectre** — Contrôle des opérations de base telles que la définition de la gamme de fréquence et le couplage de balayage. Le modèle comprend également des fonctionnalités avancées comme la configuration du marqueur et son interrogation.

## Améliorations apportées aux contrôles .NET et ActiveX (Windows)

La dépendance de thread de l'interface utilisateur n'existe plus pour les contrôles .NET. Vous n'êtes pas obligé de définir un VI ou un sous-VI à exécuter dans le thread de l'interface utilisateur pour les opérations de contrôle .NET. Par défaut, LabVIEW définit tout contrôle .NET pour qu'il s'exécute dans le thread correct, qui, dans la plupart des cas, est le thread de l'interface utilisateur.

LabVIEW ne prend pas en charge la mise au point dans les VIs Callback ActiveX et .NET. Si vous utilisez des points d'arrêt dans les VIs Callback ActiveX et .NET, LabVIEW n'effectue pas de pause au point d'arrêt.

Dans LabVIEW 8.0 et version antérieure, la sonde de refnum de .NET affiche uniquement la valeur hexadécimale de la référence. Dans LabVIEW 8.2, la Sonde Refnum .NET affiche le résultat de l'appel de la méthode ToString() sur l'objet représenté par le refnum .NET, le type de l'objet, le code de hachage de l'objet et la valeur hexa de la référence. Vous pouvez utiliser cette sonde pour définir un point d'arrêt si la valeur est un refnum non valide.

Les modifications suivantes ont été apportées aux menus .NET et ActiveX dans LabVIEW 8.2 :

- Les éléments de menu **Outils» .NET et ActiveX, Ajouter les contrôles .NET à la palette** et **Ajouter les contrôles ActiveX à la palette**, sont désormais **Outils» Importer» Contrôles .NET sur une palette** et **Outils» Importer» Contrôles ActiveX sur une palette**, respectivement.
- Le menu **Outils» .NET et ActiveX» Parcourir les propriétés ActiveX** est désormais **Affichage» Navigateur de propriétés ActiveX**.

## Améliorations apportées à l’Outil de recherche d’exemples NI

Par défaut, le dossier d’exemples **Sélections récentes** apparaît sous l’onglet **Parcourir** de l’outil de recherche d’exemples NI. Vous pouvez définir le nombre maximal d’exemples à afficher dans le dossier d’exemples **Sélections récentes** en cliquant sur le bouton **Configuration** de l’Outil de recherche d’exemples NI et en cliquant sur l’onglet **Général**.

Reportez-vous à l’*Aide de l’Outil de recherche d’exemples NI* pour obtenir de plus amples informations sur les améliorations apportées à l’Outil de recherche d’exemples NI. Cliquez sur le bouton **Aide** dans l’Outil de recherche d’exemples NI pour ouvrir l’aide.

## Améliorations apportées au contrôle de source

Reportez-vous au livre **Principes de base»Organisation et gestion d’un projet**, sur l’onglet **Sommaire** de l’*Aide LabVIEW*, pour obtenir des informations plus détaillées sur le contrôle de source dans LabVIEW.

L’intégration et les fonctionnalités de base des gestionnaires de contrôle de source de tiers suivants ont été testées avec le système de développement professionnel de LabVIEW 8.2 :

- Seapine Surround SCM
- Borland StarTeam
- Telelogic Synergy
- PushOK (plug-ins CVS et SVN)
- ionForge Evolution



**Remarque** À présent, ionForge Evolution version 2.8 ou version ultérieure fonctionne avec LabVIEW.

En outre, d’autres gestionnaires de contrôle de source tiers, qui n’étaient pas compatibles avec LabVIEW 8.0, sont désormais compatibles avec la procédure qu’utilise LabVIEW pour terminer une différenciation graphique des VIs.

## Opérations de contrôle de source sur des dossiers de projets LabVIEW

Vous pouvez effectuer des opérations de contrôle de source sur un dossier d’éléments dans un projet LabVIEW. LabVIEW effectue l’opération sur tous les éléments appropriés dans la hiérarchie. Par exemple, si vous ajoutez des fichiers au contrôle de source, LabVIEW ajoute uniquement les fichiers du dossier qui n’ont pas encore été ajoutés au contrôle de source. Lorsque vous effectuez des opérations de contrôle de source sur un dossier

d'éléments, toutes les options de configuration du contrôle de source applicables aux éléments du dossier apparaissent.



**Remarque** Si vous avez une bibliothèque de projet dans un dossier, les opérations de contrôle de source que vous effectuez ne s'appliquent pas aux éléments contenus dans la bibliothèque de projet. Pour effectuer des opérations de contrôle de source sur des éléments contenus dans une bibliothèque de projet, vous devez sélectionner les éléments manuellement. Vous pouvez effectuer des opérations de contrôle de source sur un dossier d'une bibliothèque de projet.

## Fichiers non enregistrés dans les opérations de contrôle de source

Si vous essayez d'effectuer des opérations de soumission sur des fichiers qui contiennent des modifications non enregistrées, la boîte de dialogue **Fichiers non enregistrés** vous demande d'enregistrer ou d'ignorer les modifications. Cette boîte de dialogue apparaît si vous cliquez sur le bouton **OK** dans la boîte de dialogue **Opérations de contrôle de source** et qu'un type de fichier LabVIEW compris dans l'opération de contrôle de source contient des modifications non enregistrées. Cette boîte de dialogue apparaît également si vous sélectionnez **Outils»Contrôle de source»Afficher les différences** dans une bibliothèque ou un fichier de projet qui contient des modifications non enregistrées.

## Autres améliorations apportées au contrôle de source

Les autres améliorations suivantes ont été apportées au contrôle de source dans LabVIEW 8.2 :

- Dans LabVIEW 8.0, l'élément de menu **Ajouter au contrôle de source** est disponible pour les fichiers non enregistrés dans le menu **Outils»Contrôle de source**. Dans LabVIEW 8.2, l'élément de menu **Ajouter au contrôle de source** n'est pas disponible tant que le fichier n'est pas enregistré.
- Sélectionnez **Outils»Contrôle de source»Configurer le contrôle de source** et utilisez les options **Exclure vi.lib** et **Exclure instr.lib** pour exclure les fichiers des répertoires `vi.lib` et `instr.lib` lorsque vous ajoutez des fichiers au contrôle de source. Si vous configurez LabVIEW de manière à inclure les dépendances lorsque vous ajoutez des fichiers au contrôle de source, ces options excluent les fichiers non nécessaires de l'opération.
- La boîte de dialogue **Historique des révisions dans Perforce** comprend les options **Synchroniser sur cette révision** et **Décrire la liste de modifications**. Pour accéder à ces options, cliquez avec le bouton droit sur une révision dans la liste **Historique des révisions**.

- Dans la boîte de dialogue **Historique des révisions dans Perforce**, le champ **Actions dans la liste de modifications** affiche les actions prises dans la liste de modifications actuelle. Des opérations possibles sont : ajout, édition, suppression, branchement et intégration. Ce champ affiche aussi le type de fichier sur lequel l'action se produit.
- **(Windows)** Si vous installez l'installateur Perforce, **SCM Perforce** apparaît dans le menu déroulant **Nom du gestionnaire de contrôle de source** de la page **Contrôle de source** de la boîte de dialogue **Options**.

## Améliorations apportées au format TDM

Les améliorations suivantes ont été apportées au format TDM dans LabVIEW 8.2 :

### Format de fichier TDMS en continu

LabVIEW 8.2 intègre le format de fichier TDM en continu (.tdms) pour stocker des données binaires. Le format de fichier .tdms offre des performances d'écriture accrues par rapport au format de fichier .tdm disponible dans LabVIEW 8.0 et versions antérieures. Le format de fichier .tdms fournit aussi une interface plus simple pour définir des propriétés.

Reportez-vous à la section *Vis et fonctions TDM en continu* de ce document pour obtenir des informations plus détaillées sur les fonctions et VIs TDMS.

### Propriétés TDMS et TDM définies par l'utilisateur

Vous pouvez créer et personnaliser des propriétés définies par l'utilisateur et spécifier des propriétés DAQmx pour des fichiers de mesures binaires. Utilisez les VIs Express *Écrire dans un fichier de mesures*, *Écrire des données*, *Définir des propriétés* et *Obtenir des propriétés* pour configurer des propriétés définies par l'utilisateur pour des fichiers .tdm ou .tdms.

Cliquez sur le bouton **Avancé** sur la page de configuration du VI Express *Écrire dans un fichier de mesures* pour configurer les propriétés. Affichez la page de configuration des VIs Express *Écrire des données*, *Définir des propriétés* et *Obtenir des propriétés* pour configurer leurs propriétés.

## Importation de services Web (Windows)

Vous pouvez utiliser les services Web dans LabVIEW 8.2 sans devoir gérer la complexité des services Web. Vous pouvez transformer tout service Web en une bibliothèque de projet de VIs que vous pouvez ensuite utiliser pour programmer ce service comme s'il était disponible indépendamment sur l'ordinateur local. Vous devez indiquer une URL valide pour un WSDL (Web Service Description Language). WSDL est un langage en format

XML utilisé pour décrire un service Web et sa fonctionnalité. Sélectionnez **Outils»Importer»Service Web** pour lancer un assistant qui vous guide au cours du processus d'importation des méthodes d'un service Web et de création d'une bibliothèque de VIs.



**Remarque** Vous devez avoir .NET framework, version 1.1 ou ultérieure installée pour utiliser l'assistant Importer un service Web.

Reportez-vous au livre **Principes de base»Connectivité Windows» Concepts»Importation de services Web** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur l'importation de services Web.

## Modifications apportées aux fonctions pour le code externe

Le type de données `size_t` remplace le type `int32` à certains emplacements de l'interface du gestionnaire de mémoire pour l'utilisation par le code source externe. Cette modification est compatible avec les DLL existantes.

Reportez-vous à la rubrique *Fonctions de gestion de la mémoire* de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur les fonctions de gestion de la mémoire.

National Instruments, NI, ni.com et LabVIEW sont des marques de National Instruments Corporation. Pour plus d'informations concernant les marques de National Instruments, veuillez vous référer à la partie *Terms of Use* sur le site [ni.com/legal](http://ni.com/legal). Les autres noms de produits et de sociétés mentionnés aux présentes sont les marques ou les noms de leurs propriétaires respectifs. Pour la liste des brevets protégeant les produits National Instruments, veuillez vous référer, selon le cas : à la rubrique **Aide»Brevets** de votre logiciel, au fichier `patents.txt` sur votre CD, ou à [ni.com/patents](http://ni.com/patents). Pour consulter la liste des droits d'auteur, des conditions et des exclusions et limitations de garanties concernant les composants utilisés dans USI (Xerces C++, ICU et HDF5), veuillez vous référer à `USICopyrights.chm`.