

Notes de mise à jour LabVIEW™

Mise à jour vers LabVIEW 8.5

Ces notes de mise à jour décrivent le processus de mise à jour de LabVIEW pour Windows, Mac OS et Linux vers la version 8.5, les problèmes que vous pourriez rencontrer lors de cette mise à jour, ainsi que les nouvelles fonctionnalités.

Si vous effectuez une mise à jour à partir de LabVIEW 7.1 ou d'une version antérieure à LabVIEW 8.5, National Instruments vous recommande de consulter les documents suivants en plus des présentes notes de mise à jour pour obtenir des informations supplémentaires sur les améliorations, les nouvelles fonctionnalités et les changements apportés à LabVIEW entre les versions 7.1 et 8.5.

- **Notes de mise à jour pour LabVIEW 8.0** — Les sections *Problèmes de mise à jour et de compatibilité* et *Fonctionnalités et changements dans LabVIEW 8.0* fournissent des informations importantes pour les utilisateurs souhaitant effectuer la mise à jour. Reportez-vous au site Web de National Instruments sur ni.com/frinfo et entrez l'info-code `rdfup8` pour accéder aux *Notes de mise à jour LabVIEW 8.0*.
- **Notes de mise à jour pour LabVIEW 8.2** — Les sections *Problèmes de mise à jour et de compatibilité* et *Fonctionnalités et changements dans LabVIEW 8.2* fournissent des informations importantes pour les utilisateurs souhaitant effectuer une mise à jour. Reportez-vous au site Web de National Instruments sur ni.com/frinfo et entrez l'info-code `rdfup82` pour accéder aux *Notes de mise à jour LabVIEW 8.2*.

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations supplémentaires sur les nouvelles fonctionnalités de LabVIEW 8.5, 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 8.5..... | 3 |
| Conversion des VIs..... | 3 |
| Mise à jour des modules, toolkits, drivers d'instruments et logiciels complémentaires | 4 |
| Mise à jour des autres logiciels National Instruments..... | 5 |
| Mise à jour à partir de versions précédentes de LabVIEW | 5 |
| Problèmes de mise à jour et de compatibilité | 7 |
| Mise à jour à partir de LabVIEW 8.2 | 7 |
| Mise à jour à partir de LabVIEW 8.0 | 14 |
| Mise à jour à partir de LabVIEW 7.x | 22 |
| Mise à jour à partir de LabVIEW 6.x | 44 |
| Mise à jour à partir de LabVIEW 5.x ou versions antérieures | 49 |
| Fonctionnalités et changements dans LabVIEW 8.5..... | 50 |
| Installation de LabVIEW..... | 50 |
| Documentation de LabVIEW | 50 |
| Nouveaux VIs d'exemple | 51 |
| Améliorations apportées au diagramme | 51 |
| Améliorations apportées à la face-avant..... | 52 |
| Améliorations apportées à l'environnement..... | 54 |
| Améliorations apportées aux projets LabVIEW..... | 58 |
| Améliorations apportées aux nœuds ; nouveautés et modifications apportées aux VIs et fonctions..... | 63 |
| Nouveaux événements, propriétés, méthodes et classes | 74 |
| Améliorations apportées à l'Application Builder..... | 74 |
| Améliorations apportées à MathScript LabVIEW (Windows, pas dans la version de base)..... | 79 |
| Programmation orientée objet LabVIEW..... | 86 |
| Changements des icônes de classe LabVIEW et de bibliothèque de projet..... | 88 |
| Meilleure utilisation de la mémoire virtuelle | 89 |
| Fusion de VIs..... | 90 |
| Améliorations apportées à l'Assistant Importer une bibliothèque partagée..... | 90 |
| Améliorations du contrôle d'image 3D | 91 |
| Utilisation de structures cadencées pour la programmation multicœur..... | 92 |
| Type de données en virgule fixe..... | 93 |
| Améliorations apportées au contrôle de source..... | 93 |

Mise à jour vers LabVIEW 8.5

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 8.5* 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

Vous ne pouvez pas ouvrir un VI enregistré dans une version de LabVIEW antérieure à LabVIEW 6.0 sans d'abord contacter un représentant de NI pour obtenir un logiciel de conversion afin de mettre à jour votre code aux formats de VIs compatibles avec LabVIEW 8.5. Lorsque vous ouvrez un VI enregistré dans LabVIEW 6.0 ou dans une version ultérieure, LabVIEW 8.5 convertit et compile automatiquement ce VI. Vous devez enregistrer le VI dans LabVIEW 8.5. 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.



Remarque Les VIs enregistrés dans LabVIEW 8.5 ne peuvent pas être chargés dans une version antérieure de LabVIEW. Sélectionnez **Fichier»Enregistrer pour une version précédente** pour enregistrer des VIs afin qu'ils puissent s'exécuter dans LabVIEW 8.2 ou 8.0. Avant d'enregistrer les VIs dans LabVIEW 8.5 après les avoir convertis, conservez une copie de sauvegarde des VIs que vous comptez utiliser dans LabVIEW 8.2 ou 8.0.

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 modules, toolkits, drivers d'instruments et logiciels complémentaires

Après avoir installé LabVIEW 8.5, assurez-vous que vos toolkits et logiciels supplémentaires sont compatibles avec LabVIEW 8.5, puis réinstallez-les dans le répertoire LabVIEW 8.5. 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.

Les versions des modules de LabVIEW doivent correspondre à la version de LabVIEW. Par exemple, vous devez utiliser le module LabVIEW Real-Time 8.5 avec LabVIEW 8.5. Il est possible que LabVIEW 8.5 ne prenne pas en charge les logiciels supplémentaires conçus pour des versions antérieures de LabVIEW.

Reportez-vous au site Web de National Instruments sur ni.com/fr/info et entrez l'info-code `compat` pour obtenir des informations complémentaires sur les modules et toolkits LabVIEW compatibles avec la version actuelle de LabVIEW.

Les drivers d'instruments suivants nécessitent des mises à jour ou des téléchargements pour être utilisés dans LabVIEW 8.5 :

- 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.
- Le VI Obtenir attribut instr. et le VI Définir attribut instr. ne sont plus inclus dans LabVIEW. Le driver d'instrument du Panasonic VP7723A/VP7724A Audio Analyzer utilisait ces VIs. Pour obtenir une version du driver VP7723A/VP7724A compatible avec LabVIEW 8.5, téléchargez ce driver de la base de drivers d'instruments (Instrument Driver Network) National Instruments sur ni.com/idnet.

Vous devez en outre recompiler les VIs des toolkits, drivers d'instruments et logiciels supplémentaires existants pour pouvoir les utiliser dans LabVIEW 8.5. Dans la mesure où la version du module LabVIEW doit correspondre à la version de LabVIEW, vous ne devez pas recompiler les VIs du module. Reportez-vous à la section *Conversion des VIs* du présent document pour en savoir plus sur la recompilation des VIs.

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 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.

NI TestStand 3.5 et NI TestStand 4.0 renvoient une erreur si vous essayez de configurer les VIs Express de LabVIEW 8.5 suivants :

- Fermer le stockage de données
- Formule
- Obtenir des propriétés
- Ouvrir un stockage de données
- Lire des données
- Définir des propriétés
- Mesures spectrales
- Écrire des données

Reportez-vous au site Web de National Instruments sur ni.com/frinfo et entrez l'info-code `rdtf10` pour obtenir des informations supplémentaires sur l'erreur.

Vous devez utiliser NI Spy 2.3 ou une version ultérieure avec LabVIEW 8.x. NI Spy 2.5 est disponible sur le CD de drivers de périphériques National Instruments.

Vous devez utiliser Measurement Studio 8.0 ou version ultérieure avec LabVIEW 8.5. Reportez-vous au site Web de National Instruments sur ni.com/frinfo et entrez l'info-code `exd8yy` pour accéder à l'assistant de mise à jour (Upgrade Advisor) et commander Measurement Studio 8.0.

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 8.5 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'installateur de LabVIEW 8.5 et définissez le répertoire d'installation au même répertoire `labview` que celui dans lequel vous aviez installé la version précédente de LabVIEW.

(Windows) Vous pouvez aussi remplacer la version existante de LabVIEW par LabVIEW 8.5 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 enregistré 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 8.5 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 8.5, copiez le fichier de préférences LabVIEW dans le répertoire LabVIEW 8.5.

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

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

(Linux) LabVIEW enregistre les préférences dans le fichier `.labviewrc` du 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 8.5, copiez les fichiers dans le répertoire `user.lib` du répertoire LabVIEW 8.5.

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 8.5.

Mise à jour à partir de LabVIEW 8.2

Il est possible que vous rencontriez les problèmes de compatibilité suivants lorsque vous passez de LabVIEW 8.2 à LabVIEW 8.5.



Remarque Vous pouvez aussi consulter le site Web de National Instruments sur ni.com/frinfo et entrer l'info-code `ex5d8c` pour obtenir plus d'informations sur d'éventuels problèmes que vous pouvez rencontrer lors de la mise à jour à partir de LabVIEW 8.2.x.

Plates-formes supportées

LabVIEW 8.5 comprend les changements suivants affectant les plates-formes supportées :

- LabVIEW 8.5 supporte Windows Vista x86 et x64.
- LabVIEW 8.5 est compatible avec les ordinateurs Macintosh dotés de processeurs Intel et PowerPC.

Configuration système requise

(Windows) L'installation de LabVIEW 8.5 exige un minimum de 1,2 Go d'espace disque.

(Mac OS) LabVIEW 8.5 exige un minimum de 502 Mo d'espace disque pour l'installation minimale de LabVIEW ou 734 Mo pour l'installation complète de LabVIEW.

(Linux) LabVIEW 8.5 exige un minimum de 450 Mo d'espace disque pour l'installation minimale de LabVIEW ou 640 Mo pour l'installation complète de LabVIEW.

Documentation imprimée

Les documents suivants n'ont pas été modifiés pour LabVIEW 8.5. Leur contenu ne reflète donc pas forcément les modifications apportées à LabVIEW 8.5.

- **(LabVIEW 8.2 et 8.5)** *Carte de référence rapide LabVIEW*
- **(LabVIEW 8.2 et 8.5)** *Manuel Principes de base de LabVIEW* — Dans la mesure où le *Manuel Principes de base de LabVIEW* constitue 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.

Problèmes de compatibilité avec Windows Vista

LabVIEW 8.5 supporte l'OS Windows Vista sur des systèmes 32 et 64 bits avec les changements de fonctionnalité suivants.

Les VIs Lire les registres et Écrire les registres n'apparaissent pas sur la palette **Fonctions** car ils donnent des droits de lecture et d'écriture sur n'importe quel port d'E/S du système, ce qui n'est pas recommandé pour des raisons de sécurité sur Windows Vista.

- **(Windows Vista x86)** Des composants de VIs s'installent correctement mais apparaissent comme non signés dans le journal de Windows Defender. Les VIs ne s'exécutent pas correctement.
- **(Windows Vista x64)** Ces VIs renvoient l'erreur -4850.

Modification du comportement de plusieurs VIs et fonctions

Le comportement des fonctions et VIs suivants a changé dans LabVIEW 8.5.

Améliorations apportées aux VIs et fonctions d'analyse

À chaque version de LabVIEW, National Instruments améliore de nombreux algorithmes de fonctions C et LabVIEW. National Instruments met aussi à jour LabVIEW pour qu'il utilise les derniers compilateurs. Ces améliorations, conjuguées aux changements logiciels et matériels, peuvent provoquer des différences dans les résultats numériques entre LabVIEW 8.2.x ou une version antérieure et LabVIEW 8.5. Lorsque vous comparez des nombres à virgule flottante, double précision, il se peut que vous notiez de petites différences de l'ordre de $1E-16$. Reportez-vous au site Web de National Instruments sur ni.com/frinfo en entrez l'info-code `exiigr` pour obtenir des informations complémentaires sur la comparaison des nombres à virgule flottante.

VIs Mathématiques

Les modifications suivantes ont été apportées aux **VIs de mathématiques** dans LabVIEW 8.5 :

- **Find All Zeroes of $f(x)$** — Dans la version anglaise de LabVIEW, le nom de ce VI est désormais Find All Zeros of $f(x)$.
- **Zeroes and Extrema of $f(x)$** — Dans la version anglaise de LabVIEW, ce VI a été renommé Zeros and Extrema of $f(x)$.

Fonctions numériques

Dans LabVIEW 8.5, les modifications suivantes ont été apportées aux fonctions **numériques** :

- **Arrondir vers +l'infini** — La fonction Arrondir à +l'infini a été renommée Arrondir vers +l'infini
- **Arrondir vers -l'infini** — La fonction Arrondir à -l'infini a été renommée Arrondir vers -l'infini

VIs de traitement de signaux

Dans le VI Mesures de transition, la sortie **preshoot** est devenue **avant transition**. Cette sortie a également changé de type : elle est passée du type de données numérique à virgule flottante double précision 64 bits à un type de données cluster. La sortie **overshoot** est devenue **après transition**. Cette sortie a également changé de type : elle est passée du type de données numérique à virgule flottante double précision 64 bits à un type de données cluster.

Fonctions hyperboliques

Dans LabVIEW 8.5, les modifications suivantes ont été apportées aux fonctions hyperboliques suivantes :

- La fonction Arc cosinus hyperbolique renvoie NaN quand la valeur de l'entrée est un nombre réel qui se trouve hors de la gamme de la fonction.
- La fonction Arc cosinus hyperbolique renvoie NaN quand la valeur en entrée est un nombre réel qui se trouve hors de la gamme de la fonction.

VIs et fonctions de bibliothèques et exécutables

Dans la fonction Appeler une fonction d'une DLL, lorsque vous configurez un pointeur de chaîne Pascal, vous devez câbler une valeur à l'entrée chaîne sur le diagramme. Lorsque vous configurez un pointeur de chaîne C, vous devez câbler une valeur à l'entrée chaîne ou spécifier la taille de la chaîne dans le menu déroulant **Taille minimale** de l'onglet **Paramètres** de la

boîte de dialogue **Appeler une fonction d'une DLL**. Vous ne pouvez pas exécuter le VI tant que vous n'avez pas spécifié de valeurs pour les chaînes.

Fonction Ouvrir une référence de VI

Dans LabVIEW 8.0.x et 8.2.x si le nom du VI à l'entrée **Chemin du VI** correspond au nom d'un VI en mémoire sur cette cible, LabVIEW renvoie une référence au VI en mémoire et ne charge pas le VI spécifié à l'entrée **Chemin du VI**. Dans LabVIEW 8.5, si le nom du VI à l'entrée **Chemin du VI** correspond au nom d'un VI en mémoire sur cette cible mais que le chemin diffère, la fonction Ouvrir une référence de VI renvoie une erreur.

Terminaux de VIs polymorphes qui supportent les types de données 64 bits et numériques double précision

LabVIEW contraint les données numériques à précision étendue à des données numériques double précision si vous les câblez au terminal d'un VI polymorphe qui supporte les types de données numériques double précision et entiers 64 bits. Cette coercition préserve une portion du composant fractionnaire des données d'origine.

Modification diverses du comportement de certains VIs et fonctions

D'autres modifications, décrites ci-après, ont été apportées aux fonctions et VIs de LabVIEW 8.5 :

- Le VI Obtenir attribut instr. et le VI Définir attribut instr. ne sont plus inclus dans LabVIEW. Si vous utilisez un ou plusieurs de ces VIs dans une application, remplacez-les par le nœud de propriété sur la palette **VISA avancé** pour obtenir une fonctionnalité équivalente.
- Le paramètre **Tous les dossiers** du VI Liste récursive des fichiers peut contenir des raccourcis vers des dossiers mais le VI n'effectue pas de récursion.

Modification du comportement de propriétés, méthodes et événements

Le comportement des propriétés, méthodes et événements suivants a changé dans LabVIEW 8.5 :

- La propriété Liaison des données : chemin de la classe Commande est en lecture/écriture et peut être définie pendant l'exécution du VI. Pour écrire cette propriété, vous devez lier la commande à l'URL NI Publish-Subscribe-Protocol avant de commencer à écrire.
- La propriété cible : système d'exploitation de la classe Application inclut les valeurs `Windows x64` et `Linux x64`.

- La méthode Point en ligne-colonne de la classe CommandeArbre renvoie le tag `TREE_COLUMN_HEADERS` lorsque vous câblez un point dans l'en-tête de colonne de l'arborescence.
- La méthode classe LabVIEW : créer inclut une entrée nom. Si vous ne câblez pas l'entrée **nom**, LabVIEW demande à l'utilisateur de nommer la classe à l'exécution.
- Les méthodes Valeur de commande : obtenir [variant], Valeur de commande : obtenir [aplatie], Valeur de commande : définir [variant] et Valeur de commande : définir [aplatie] ne suppriment plus les espaces blancs au début et à la fin lors de la recherche de commandes.

Propriétés, méthodes et événements obsolètes

LabVIEW 8.5 ne supporte pas les propriétés, méthodes et événements suivants :

- Propriété Instance par défaut de la classe BibliothèqueClasseLV. À la place, utilisez le VI Obtenir la valeur par défaut de classe LabVIEW.
- Propriété Géométrie de la classe ObjetScène. À la place, utilisez la propriété Traçable.
- Propriété Couleurs de grille de la classe GrapheEtGrapheDéroulant. À la place, utilisez la propriété Couleurs de grille.
- Propriété Couleurs de grille : couleur X de la classe GrapheEtGrapheDéroulant. À la place, utilisez les propriétés Couleurs de grille : couleur majeure et Couleurs de grille : couleur mineure.
- Propriété Couleurs de grille : couleur Y de la classe GrapheEtGrapheDéroulant. À la place, utilisez les propriétés Couleurs de grille : couleur majeure et Couleurs de grille : couleur mineure.
- Propriété Légendes : tracés affichés de la classe GrapheDéroulant. À la place, utilisez la propriété Légende : nombre de lignes.
- Propriété Légendes : tracés affichés de la classe Graphe. À la place, utilisez la propriété Légende : nombre de lignes.
- Propriété Largeur de pixel de la classe ListeDéroulante. À la place, utilisez la propriété Limites : largeur.
- Propriété Barres de défilement visibles de la classe Image. À la place, utilisez les propriétés Barre de défilement horizontale visible et Barre de défilement verticale visible.
- Méthode Géométrie de la classe ObjetScène. À la place, utilisez la méthode Définir un attribut traçable.
- Méthode Scène : géométrie : nouveau maillage de la classe Application. À la place, utilisez la méthode Scène : traçable : géométrie : nouveau maillage.

- Événement Début de glisser de la classe Commande. À la place, utilisez l'événement Début de glisser de la classe de commande appropriée.
- Événement Début de glisser ? de la classe Commande. À la place, utilisez l'événement Début de glisser ? de la classe de commande appropriée.

Propriétés, méthodes et événements renommés

Les propriétés, méthodes et événements suivants ont été renommés dans LabVIEW 8.5 :

| Classe | Nom dans LabVIEW 8.2 | Nom dans LabVIEW 8.5 | Type |
|------------------------|--|--|-----------|
| TempsAbsolu, Numérique | Gamme des données | Limites d'entrée des données | Propriété |
| TempsAbsolu, Numérique | Gamme des données : incrément | Limites d'entrée des données : incrément | Propriété |
| TempsAbsolu, Numérique | Gamme des données : maximum | Limites d'entrée des données : maximum | Propriété |
| TempsAbsolu, Numérique | Gamme des données : minimum | Limites d'entrée des données : minimum | Propriété |
| TempsAbsolu, Numérique | Action hors gamme | Réponse aux valeurs hors gamme | Propriété |
| TempsAbsolu, Numérique | Action hors gamme : incrément | Réponse aux valeurs hors gamme : incrément | Propriété |
| TempsAbsolu, Numérique | Action hors gamme : maximum | Réponse aux valeurs hors gamme : maximum | Propriété |
| TempsAbsolu, Numérique | Action hors gamme : minimum | Réponse aux valeurs hors gamme : minimum | Propriété |
| Application | Bibliothèque : obtenir la version du fichier de bibliothèque de projet | Bibliothèque : obtenir la version LabVIEW d'un fichier | Méthode |
| Application | Scène : géométrie : nouvelle boîte | Scène : traçable: géométrie : nouvelle boîte | Méthode |
| Application | Scène : géométrie : nouveau cône | Scène : traçable: géométrie : nouveau cône | Méthode |
| Application | Scène : géométrie : nouveau cylindre | Scène : traçable: géométrie : nouveau cylindre | Méthode |

| Classe | Nom dans LabVIEW 8.2 | Nom dans LabVIEW 8.5 | Type |
|---------------------------------------|---|--|-----------|
| Application | Scène : géométrie : nouvelle carte de niveaux | Scène : traçable : géométrie : nouvelle carte de niveaux | Méthode |
| Application | Scène : géométrie : nouvelle grille | Scène : traçable: géométrie : nouveau maillage | Méthode |
| Application | Scène : géométrie : nouvelle sphère | Scène : traçable: géométrie : nouvelle sphère | Méthode |
| Application (ActiveX) | LibraryGetProjectLibFileVersion | LibraryGetFileLVVersion | Méthode |
| Numérique, Texte numérique et Échelle | Format et précision | Format d'affichage | Propriété |
| Numérique, Texte numérique et Échelle | Format et précision : format | Format d'affichage : format | Propriété |
| Numérique, Texte numérique et Échelle | Format et précision : précision | Format d'affichage : précision | Propriété |
| TableNumériques | En-têtes de colonnes visibles | Numéro de signal visible | Propriété |
| TableNumériques | En-têtes de lignes visibles | Transitions visibles | Propriété |
| AffichageGrapheScène et FenêtreScène | Couleur de transparence | Couleur de l'arrière-plan | Propriété |
| ObjetScène | Définir la géométrie | Définir un attribut traçable | Méthode |
| VI | Connecteur | Connecteur : définir | Propriété |

Modification du comportement de MathScript LabVIEW (Windows, pas dans la version de base)

Les modifications supplémentaires suivantes ont été apportées à MathScript LabVIEW dans LabVIEW 8.5 :

- Les changements que vous effectuez dans la liste des chemins de la recherche ou dans le répertoire de travail en utilisant les fonctions MathScript suivantes s'appliquent uniquement à l'instance actuelle de la **fenêtre MathScript LabVIEW** ou le nœud MathScript à partir duquel vous appelez la fonction :
 - `addpath`
 - `cd`
 - `path`
 - `rmpath`

LabVIEW réinitialise la liste des chemins de recherche à la liste par défaut lorsque vous fermez la **fenêtre MathScript LabVIEW** ou lorsque le VI qui contient le nœud MathScript arrête de s'exécuter.

- La syntaxe de la fonction qz est passée de [q, z, alpha, beta, evec] = qz(a, b) à [S, T, Q, Z, R, L] = qz(A, B, type).

Icônes de classe LabVIEW

Si vous avez créé une icône de classe LabVIEW dans LabVIEW 8.2 et que vous voulez qu'elle s'affiche lorsque vous placez une commande ou un indicateur de classe sur le diagramme, vous devez mettre l'icône de classe à jour pour qu'elle occupe moins d'espace et qu'elle ne cache pas une partie de l'icône de classe. Utilisez une image ne dépassant pas 32 pixels de large sur 19 pixels de haut.

Ouverture de LLBs dans LabVIEW

L'option **Support des LLBs dans l'Explorateur Windows** sur la page **Environnement** de la boîte de dialogue **Options** n'existe plus. LabVIEW ouvre des LLBs dans la fenêtre **Gestionnaire de LLBs**. Reportez-vous au site Web de National Instruments sur ni.com/frinfo et entrez l'info-code `exvfc5` pour obtenir davantage d'informations sur l'ouverture de LLBs.

Restriction du niveau de priorité des boucles cadencées

Dans LabVIEW 8.2.x et version antérieure, vous pouvez sélectionner jusqu'à 2 exposant 32 pour le niveau de priorité d'une boucle cadencée. LabVIEW 8.5 supporte uniquement des niveaux de priorité inférieurs à 65 535.

Type de données waveform

Lorsque l'indexation dépasse les limites d'un tableau de waveforms, la waveform résultante est une waveform par défaut valide avec une valeur dt égale à 1 plutôt qu'une waveform non valide ayant une valeur dt égale à 0. Ceci est également vrai en cas d'exécution zéro fois d'une boucle For avec un tunnel de sortie scalaire.

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 8.5. Reportez-vous à la section *Mise à jour à partir de LabVIEW 8.2* 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 8.0 et 8.5, sur ni.com/manuals, pour obtenir des informations complémentaires sur les nouvelles fonctionnalités et modifications apportées à ces versions.

Plates-formes supportées

Les changements suivants ont été apportés à LabVIEW 8.2 et versions ultérieures en matière de support de plates-formes :

- LabVIEW 8.2 et versions ultérieures ne supportent pas Windows XP x64.
- LabVIEW 8.2 et versions ultérieures ne supportent 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/fr/info et entrez l'info-code `macintel` pour obtenir davantage d'informations sur le support pour Macintosh. LabVIEW 8.5 supporte les ordinateurs Macintosh dotés de processeurs Intel et PowerPC.

Configuration système requise

(Windows) LabVIEW 8.2 et versions ultérieures exigent un minimum de 1,2 Go pour l'installation de LabVIEW.

(Mac OS) LabVIEW 8.2 exige un minimum de 500 Mo d'espace disque pour l'installation minimale de LabVIEW ou 700 Mo pour l'installation complète de LabVIEW. LabVIEW 8.5 exige un minimum de 502 Mo d'espace disque pour l'installation minimale de LabVIEW ou 734 Mo pour l'installation complète de LabVIEW.

(Linux) LabVIEW 8.2 exige un minimum de 430 Mo d'espace disque pour l'installation minimale de LabVIEW ou 620 Mo pour l'installation complète. LabVIEW 8.5 exige un minimum de 450 Mo d'espace disque pour l'installation minimale de LabVIEW ou 640 Mo pour l'installation complète de LabVIEW.

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 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 et versions ultérieures.

Communication entre plusieurs instances d'application

Dans LabVIEW 8.2 et versions ultérieures, 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 **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 les versions 8.0 et 8.2, ou ultérieure, de LabVIEW sont installées 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 ou une version ultérieure.

VIs d'E/S sur fichiers

Les VIs Écrire dans un fichier tableur et Lire un fichier tableur sont des VIs 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 ou version ultérieure, 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-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.

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 coefficients 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 pas 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 et versions ultérieures, 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 et versions ultérieures, 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

Dans LabVIEW 8.2 et version ultérieure, les modifications suivantes ont été apportées aux VIs Waveform :

- **VI Détection élémentaire du niveau de déclenchement** — Dans les deux instances de ce VI, l'entrée **pen**te s'appelle désormais **pen**te 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 et versions ultérieures :

- 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. auto.** 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. Dans LabVIEW 8.2 et versions ultérieures, les méthodes .NET qui transmettent des types de données tableau par référence transmettent les données avec le type de données réel.
- 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 et versions ultérieures, 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 et versions ultérieures ne supportent pas les propriétés, méthodes et événements suivants.

- LabVIEW 8.2 et versions ultérieures ne supportent 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 et versions ultérieures :

| Classe | Nom dans LabVIEW 8.0 | Noms dans LabVIEW 8.2 et versions ultérieures | 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'application ? | Événement |
| GrapheIntensité, GrapheSignauxMixte et Graphe | Palette de curseur : visible | Légende des curseurs visible | Propriété |
| Bibliothèque | Supprimer un tag de bibliothèque | Tag de bibliothèque : supprimer | 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 |
| 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 |

| Classe | Nom dans LabVIEW 8.0 | Noms dans LabVIEW 8.2 et versions ultérieures | Type |
|---|--|---|-----------|
| ListeDéroulante, ListeMultiColonne et CommandeArbre | Glisser/Déposer : autoriser le glissement d'éléments | Glisser/Déposer : autoriser le glissement | 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 |

Modifications apportées à l'Application Builder

Dans LabVIEW 8.2 et version ultérieure, vous ne pouvez pas afficher le contenu d'une application autonome (EXE) ou d'une bibliothèque partagée (DLL) en renommant l'application ou la bibliothèque partagée avec l'extension de fichier .11b. Vous ne pouvez pas non plus accéder à un VI d'une application autonome ou d'une bibliothèque partagée si vous spécifiez le chemin du VI hors de l'application ou de la bibliothèque partagée. Reportez-vous au site Web de National Instruments sur ni.com/frinfo et entrez l'info-code exjk3b pour en savoir plus sur l'affichage et l'accès à des applications et des bibliothèques partagées.

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 8.5. Reportez-vous aux sections *Mise à jour à partir de LabVIEW 8.0* et *Mise à jour à partir de LabVIEW 8.2* 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 8.5, sur 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 Principes de base de LabVIEW* n'ont pas été modifiés pour LabVIEW 8.2 ou 8.5. 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 exige un minimum de 256 Mo de RAM, mais National Instruments recommande 1 Go 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 8.5 exige un minimum de 502 Mo d'espace disque pour l'installation minimale de LabVIEW ou 734 Mo pour l'installation complète de LabVIEW.

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 8.5 exige un minimum de 450 Mo d'espace disque pour l'installation minimale de LabVIEW ou 640 Mo pour l'installation complète de LabVIEW.

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/fr/info 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 en 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 des éléments de spécification du format %o, %b 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 chargez un VI créé dans 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 matriciel et qui utilise un tableau 2D à la place, un symbole 7.x rouge apparaît sur la fonction. Le symbole rouge indique que LabVIEW a remplacé le tableau 2D par un type de données matriciel.

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 essaie alors de charger le VI à partir du disque, à l'emplacement spécifié. Dans LabVIEW 8.5, une erreur se produit si LabVIEW ne trouve pas le fichier ou si le nom de VI du fichier est le même que le nom de VI qualifié d'un autre VI en mémoire sur cette cible.

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 l'instance chaîne du 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, l'instance chaîne du 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 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 met à jour le VI afin de maintenir la fonctionnalité 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 Réel/Im en polaire et Polaire en Réel/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** (l'équivalent pour le VI Mesures de transition est la sortie **après transition**), ou **preshoot** (l'équivalent pour le VI Mesures de transition est la sortie **avant transition**) ; utilisez 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 d'état haut** ou **niveau d'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.

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. auto.** de la méthode Exécuter le VI à VRAI, LabVIEW supprime automatiquement la référence à la fin de l'exécution du VI. Si la méthode Exécuter le VI génère une erreur, la référence est automatiquement fermée. Dans LabVIEW 8.0, LabVIEW supprime automatiquement 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. Dans LabVIEW 8.2 et versions ultérieures, ce comportement est repassé à celui de 7.1.

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 7.x 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 ne sera plus 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 un pixel de moins pour chaque ligne de la commande arbre par rapport à LabVIEW 8.x. Par exemple, si vous chargez un VI LabVIEW 7.x dans LabVIEW 8.x qui contient un nœud de propriété qui renvoie la hauteur et la largeur d'une commande arbre de 70 pixels et 16 pixels, tout nouveau nœud de propriété que vous placez pour définir la hauteur et la largeur renvoient 69 pixels et 16 pixels.

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, ListeMultiColonne, 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 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 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 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.

Dans LabVIEW 8.x, vous ne pouvez pas sélectionner les événements Souris appuyée, Souris appuyée ?, Déplacement de la souris ou Souris relâchée de la classe VI dans la boîte de dialogue **Éditer les événements**. À la place, utilisez respectivement les événements Souris appuyée, Souris appuyée ?, Déplacement de la souris et Souris relâchée de la classe Panneau.

Tags des éléments de l'application

La liste suivante comprend les tags d'éléments de l'application qui ont été supprimés de LabVIEW soit parce que cette fonctionnalité n'est plus disponible, soit parce qu'elle a été combinée avec d'autres fonctionnalités :

- APP_SAVE_WITH_OPTIONS
- APP_UPDATE_VXI
- APP_DSC_TOOLBAR
- APP_DSC_TAGEDITOR
- APP_DSC_TAGMONITOR
- APP_DSC_HTV
- APP_DSC_ENGINE
- APP_DSC_SECURITY
- APP_DSC_LOGOUT
- APP_DSC_CPWD
- APP_DSC_USERINFO
- APP_DSC_USEREDITOR
- APP_DSC_ADVANCED

- APP_DSC_STARTUP
- APP_DSC_SRVBRW
- APP_DSC_IIST
- APP_DSC_IMAGENAV
- APP_DSC_OPTIONS
- APP_SRC_CODE_CTRL
- APP_BUILD_STANDALONE_APP
- APP_EDIT_VI_LIBRARY
- APP_DN_ASSEMBLY_REFS
- APP_SHOW_CLIPBOARD
- APP_VIEW_PRINTED_MANUALS
- APP_RT_ENGINE_INFO
- APP_SWITCH_EXEC_TARGET
- APP_REALTIME

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 Aplatir 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 à partir de 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 Aplatir 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 Aplatir 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 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 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 -32 768, 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 32 767, 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 8.5. Reportez-vous aux sections *Mise à jour à partir de LabVIEW 7.x*, *Mise à jour à partir de LabVIEW 8.0* et *Mise à jour à partir de LabVIEW 8.2* 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 8.5, sur 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 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 série.

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\smp_lser1.lib` 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. 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 sous la version 6.1 | Définition sous 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-19 J | 1,60217642e-19 J |
| lumen/pied carré | 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 |
| 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 Aplatir en chaîne pour aplatir 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 ^ 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 ^ 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 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 sur ni.com/frinfo et entrez l'info-code `ext8h9` pour savoir comment effectuer une mise à niveau de LabVIEW 5.x ou version antérieure, vers LabVIEW 8.5.

Fonctionnalités et changements dans LabVIEW 8.5

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 8.5. 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 8.5.

Installation de LabVIEW

LabVIEW 8.5 offre les options d'installation suivantes sous Windows.

- **DVD** — Inclut LabVIEW, les drivers de périphériques et LabVIEW SignalExpress.
- **CD** — Inclut LabVIEW uniquement. Si vous installez LabVIEW à partir du CD, vous installez les drivers de périphériques et SignalExpress à l'aide du CD de drivers de périphériques National Instruments.

MAX (Measurement & Automation Explorer) n'apparaît plus dans l'arborescence pendant l'installation si vous utilisez le CD d'installation. Les composants de MAX requis par LabVIEW sont installés automatiquement. Vous pouvez installer MAX à partir du CD de drivers de périphériques National Instruments.

Documentation de LabVIEW

Les améliorations suivantes ont été apportées à la documentation de LabVIEW 8.5 :

- Le fichier `readme.html` comprend une liste partielle des bugs rapportés par les utilisateurs qui ont été corrigés dans la version actuelle de LabVIEW. Reportez-vous à la section *Bugs corrigés* du fichier `readme.html` qui figure dans le répertoire `labview` pour consulter la liste des bugs.
- Les rubriques de référence de l'*Aide LabVIEW* contiennent des informations spécifiques à chaque module LabVIEW si ce module est installé et s'il existe des informations spécifiques à ce module sur les VIs, fonctions, etc.

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 diagramme

Le diagramme de LabVIEW 8.5 et les fonctionnalités associées présentent les améliorations suivantes.

Création de VIs réentrants partagés

Lorsque vous configurez un VI réentrant dans LabVIEW 8.2.x et versions antérieures, LabVIEW préalloue une copie du VI réentrant en mémoire pour chaque appel à ce VI réentrant. Dans LabVIEW 8.5, vous pouvez configurer un VI réentrant pour partager le VI clone entre les appelants sans préallouer de VI dupliqué en mémoire pour chaque appel au VI réentrant. En partageant des VIs clones, vous diminuez l'utilisation de mémoire car LabVIEW ne préalloue pas de clone pour chaque appel au VI réentrant.

Pour configurer un VI afin qu'il partage le VI dupliqué entre d'autres appelants, sélectionnez **Fichier»Propriétés du VI** puis **Exécution** dans le menu déroulant **Catégorie** pour afficher la page **Propriétés de l'exécution** de la boîte de dialogue **Propriétés du VI**. Cochez l'option **Exécution réentrante** et sélectionnez l'option **Partager les copies entre instances** pour partager des VIs dupliqués.

Cependant, le partage de VIs clones risque de diminuer la vitesse d'exécution. Ne sélectionnez pas l'option **Partager les copies entre instances** si vous voulez conserver les informations d'état sur plusieurs appels d'un VI clone, comme des registres à décalage non initialisés. Pour préserver les informations d'état entre les appels à un VI dupliqué, sélectionnez l'option **Préallouer une copie pour chaque instance** sur la page **Propriétés de l'exécution**.

Améliorations apportées aux structures à conditions désactivées et structures à diagrammes désactivés

Lorsque LabVIEW charge un VI avec des objets définis par l'utilisateur, comme des sous-VIs et des définitions de type, dans le sous-diagramme Désactivé d'une structure à diagrammes désactivés ou dans les sous-diagrammes désactivés d'une structure à conditions désactivées, LabVIEW ne charge pas ces objets en mémoire. Toutefois, lorsque vous affichez le diagramme du VI, si LabVIEW ne trouve pas certains objets, ils apparaîtront avec une icône point d'exclamation. Le VI n'est pas brisé car LabVIEW n'intègre pas le code quand il compile et exécute le VI.

Améliorations apportées à la face-avant

La face-avant de LabVIEW 8.5 et ses fonctionnalités associées présentent les améliorations suivantes.

Améliorations apportées au graphe numérique

L'affichage par défaut de la légende des tracés du graphe numérique est l'affichage en arbre, qui est situé à gauche de la zone de tracé du graphe. Par défaut, l'affichage en arbre affiche à la fois les bus et les lignes numériques. Vous pouvez étendre et réduire des bus numériques dans l'affichage en arbre de la légende des tracés en cliquant sur le symbole d'extension/réduction à gauche du bus numérique. Vous pouvez utiliser l'affichage standard de la légende des tracés pour obtenir l'apparence de la légende des tracés telle qu'elle apparaît dans LabVIEW 8.2 et versions antérieures.

Par défaut, le graphe numérique affiche les lignes et les bus numériques dans la zone de tracé. Personnalisez la zone de tracé du graphe numérique pour n'afficher que les données qui vous intéressent.

Exporter une image simplifiée

L'option **Exporter une image simplifiée** sur le menu local des graphes, des graphes déroulants, des tables, des commandes image, des données numériques et des commandes waveforms numériques ne se trouve plus sur le menu local **Opérations sur les données** mais sur le menu local principal.

Vous pouvez aussi cliquer avec le bouton droit sur un graphe, un graphe déroulant, un tableau, une commande image, des données numériques ou une commande ou un indicateur numérique dans une commandeX et sélectionner **Exporter une image simplifiée** dans le menu local pour exporter une image dans le presse-papiers ou pour enregistrer une image sur disque.

Les images de la face-avant exportées à l'aide de l'élément de menu **Exporter une image simplifiée** ne s'exportent plus avec une bordure autour de l'image.

Amélioration de la légende des tracés

Personnalisez l'apparence d'un tracé dans la zone de tracé d'un graphe ou d'un graphe déroulant en cliquant sur le symbole qui correspond à ce tracé dans la légende des tracés et en sélectionnant l'une des options du menu local.

Vous pouvez ajouter une barre de défilement horizontale ou verticale à la légende de tracés des graphes et des graphes déroulants. Utilisez une barre

de défilement pour afficher des tracés dans une légende de tracés sans devoir afficher tous les tracés à la fois.

Pour ajouter une barre de défilement à une légende de tracés, cliquez avec le bouton droit sur la légende et sélectionnez **Éléments visibles»Barre de défilement horizontale** ou **Éléments visibles»Barre de défilement verticale** dans le menu local.

Reportez-vous à la rubrique *Personnalisation des graphes et des graphes déroulants* du livre **Principes de base»Graphes et graphes déroulants»Concepts** sous l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur les légendes de tracés et les barres de défilement des légendes de tracés.

Utilisation d'une case à cocher mixte

Utilisez une case à cocher mixte, disponible sur la palette **Système**, pour afficher la valeur VRAI, FAUX ou MIXTE. Par exemple, utilisez une case à cocher mixte si vous voulez afficher une série de valeurs booléennes dans le même indicateur et que ces valeurs booléennes sont toutes VRAI, toutes FAUX, ou une combinaison de VRAI et de FAUX, appelée MIXTE. Utilisez la commande Case à cocher mixte pour changer simultanément une série de valeurs booléennes à VRAI ou FAUX.

Bien que la case à cocher mixte soit une commande de type énumération, vous pouvez la configurer comme une commande booléenne.

Pour désactiver la possibilité de définir la valeur MIXTE d'une case à cocher mixte de manière interactive, cliquez avec le bouton droit sur la commande et sélectionnez **Autoriser mixte** dans le menu local pour supprimer la coche à côté de l'élément de menu. Si vous désactivez **Autoriser mixte**, LabVIEW peut assigner la valeur de la case à cocher mixte à MIXTE mais vous ne pouvez pas la changer à MIXTE à l'exécution en cliquant dessus.

Reportez-vous à la rubrique *Personnalisation des graphes et des graphes déroulants* du livre **Principes de base»Graphes et graphes déroulants** sous l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur les graphes numériques.

Autres améliorations apportées à la face-avant

Vous pouvez configurer une commande de la face-avant pour qu'elle se mette à l'échelle automatiquement pendant le redimensionnement de la face-avant. Cliquez avec le bouton droit sur une barre de division et sélectionnez **Position du panneau»Mettre les objets à l'échelle au redimensionnement** dans le menu local.

Améliorations apportées à l'environnement

Les améliorations suivantes ont été apportées à l'environnement de LabVIEW 8.5.

Enregistrement automatique pour améliorer la récupération

En cas d'arrêt anormal ou de panne système, LabVIEW 8.5 sauvegarde désormais, dans un emplacement temporaire, tous les projets modifiés (.lvproj), bibliothèque de projet (.lvlib), commandeX (.xctl) et classe LabVIEW (.lvclass) qui étaient ouverts au moment de la panne ou de l'arrêt. Ces types de fichiers apparaissent avec les fichiers de VI (.vi), VI modèle (.vit), commande (.ctl) et commande modèle (.ctt) dans la fenêtre **Sélectionner les fichiers à récupérer** la prochaine fois que vous lancez LabVIEW. Pour activer le bouton **Comparer la sauvegarde à l'original** dans la fenêtre **Sélectionner les fichiers à récupérer** pour des projets, des bibliothèques de projet, des commandeX et des fichiers de classe LabVIEW, vous devez d'abord manuellement configurer la comparaison de texte pour la récupération de fichiers.

Enregistrement pour plusieurs versions antérieures de LabVIEW

Vous pouvez enregistrer des VIs, des projets et des bibliothèques de projet au format d'une version antérieure de LabVIEW afin de faciliter la mise à jour de LabVIEW et de vous aider à maintenir des fichiers dans plusieurs versions de LabVIEW si nécessaire. Sélectionnez **Fichier»Enregistrer pour une version précédente** afin d'ouvrir la boîte de dialogue **Enregistrer pour une version précédente**. Sélectionnez **8.5**, **8.2** ou **8.0** dans le menu déroulant **Version de LabVIEW** pour enregistrer les données pour une version précédente de LabVIEW.

Améliorations apportées aux boîtes de dialogue

Les améliorations suivantes ont été apportées aux boîtes de dialogue de LabVIEW 8.5 :

Améliorations apportées à la boîte de dialogue **Ajouter des cibles et des périphériques**

Si vous ajoutez une cible dans un projet LabVIEW et qu'elle crée des conflits avec une cible existante, la boîte de dialogue **Ajouter des cibles et des périphériques** affiche la colonne **Options** à côté de la liste **Cibles et périphériques**. Cliquez sur la flèche vers le bas à côté de la résolution de conflit dans la colonne **Options** et sélectionnez une des options disponibles pour résoudre le conflit. Vous ne pourrez pas ajouter de cible tant que vous

n'aurez pas résolu tous les conflits se trouvant dans la colonne **Options** pour la cible sélectionnée.

Boîte de dialogue Appeler une fonction d'une DLL

Cliquez avec le bouton droit sur une fonction Appeler une fonction d'une DLL et sélectionnez **Configurer** dans le menu local pour afficher la boîte de dialogue **Appeler une fonction d'une DLL**. La page **Détection d'erreur** fournit plusieurs options de détection d'erreur intégrée pour la fonction Appeler une fonction d'une DLL.

- Les commandes **Maximum** et **Par défaut** de la page **Détection d'erreur** permettent à LabVIEW de récupérer des exceptions non gérées qui surviennent à la configuration de la fonction ou à l'appel d'une bibliothèque partagée ou d'une DLL.
- La commande **Désactivé** de la page **Détection d'erreur** désactive la détection d'erreur mais améliore la vitesse d'exécution de la fonction Appeler une fonction d'une DLL. Toutefois, certaines erreurs peuvent provoquer un arrêt anormal de LabVIEW. Avant de désactiver la détection d'erreur, assurez-vous que la fonction référencée par Appeler une fonction d'une DLL ne donne pas lieu à des exceptions non gérées.

La boîte de dialogue **Appeler une fonction d'une DLL** comprend également les améliorations suivantes :

- Sur la page **Paramètres**, la commande **Taille minimale** affiche **<Aucun>** par défaut. La commande **Taille minimale** est visible quand vous sélectionnez **Chaîne** dans le menu déroulant **Type** puis **Pointeur de chaîne C** dans le menu déroulant **Format de la chaîne** ou quand vous sélectionnez **Tableau** dans le menu déroulant **Type** et que vous sélectionnez **Pointeur sur les données du tableau** dans le menu déroulant **Format du tableau**.
- Sur la page **Fonction**, la commande **Réentrant** a été renommée **Exécuter dans n'importe quel thread**.

Améliorations et modifications apportées à la boîte de dialogue Options

Les améliorations suivantes ont été apportées à la boîte de dialogue **Options** de LabVIEW 8.5 :

- Sur la page **Diagramme**, l'option **Étiquettes de nom transparentes** est activée par défaut.
- Sur la page **Environnement**, l'option **Support des LLBs dans l'Explorateur Windows** n'existe plus.
- Sur la page **Face-avant**, les éléments sont classés dans l'ordre des éléments qui s'appliquent au comportement de temps d'édition et des éléments qui s'appliquent au comportement de temps d'exécution.

- Sur la page **Face-avant**, cochez l'option **Les terminaux du connecteur sont Requis par défaut** pour définir les nouveaux terminaux des connecteurs à Requis.
- Sur la page **Nouveautés et changements dans LabVIEW 8.x**, vous trouverez un lien vers l'*Aide LabVIEW : Liste complète des nouveautés et changements apportés à LabVIEW 8.x*. La rubrique *Liste complète des nouveautés et changements dans LabVIEW 8.x* comprend les nouvelles fonctionnalités et les changements entre LabVIEW 7.1 et 8.5. La page **Nouveautés et changements dans LabVIEW 8.x** inclut désormais uniquement les éléments ajoutés ou modifiés depuis la version précédente de LabVIEW.
- Sur la page **Impression**, la valeur d'impression par défaut pour la plate-forme MacIntel est standard. Avec l'exception des **Marges**, toutes les autres options d'impression pour MacIntel n'existent plus.

Améliorations et modifications apportées à la boîte de dialogue Propriétés

LabVIEW 8.5 inclut une nouvelle boîte de dialogue **Propriétés du nœud numérique**. Utilisez cette boîte de dialogue pour configurer les paramètres de sortie pour les fonctions de type **Numérique**, comme le type de données de la valeur en sortie. Si le type de données de la valeur en sortie est en virgule fixe, vous pouvez aussi configurer comment la fonction traite les conditions de dépassement et de quantification.

LabVIEW 8.5 inclut aussi les améliorations suivantes pour les pages des boîtes de dialogue **Propriétés** des commandes, des indicateurs et des constantes :

- La page **Format d'affichage** remplace la page **Format et précision**.
- Les pages **Entrée de données** et **Type de données** remplacent l'onglet **Gamme des données**.
- La section **Représentation** de la page **Type de données** comporte l'option **FXP (virgule fixe)**.
- La page **Type de données** comporte des options **Configuration à virgule fixe**.
- L'option **Utiliser les limites par défaut** remplace l'option **Utiliser la gamme par défaut** sur la page **Entrée de données**.
- La section **Taille** sur la page **Apparence** comprend la hauteur et la largeur, en pixels, des commandes et indicateurs.

Autres améliorations apportées aux boîtes de dialogue

Les améliorations supplémentaires suivantes ont été apportées aux boîtes de dialogue de LabVIEW 8.5 :

- Sur la seconde page de l'Assistant **Importer un service Web**, vous pouvez entrer des informations relatives à l'authentification ou au serveur proxy du service Web.
- La page **Entrez la bibliothèque du projet et le répertoire de destination** remplace la page **Entrez les informations sur la bibliothèque du projet et le répertoire de destination** de l'Assistant **Importer un service Web**.
- Sur la page **Paramètres généraux** des boîtes de dialogue **Propriétés de la bibliothèque de projet**, **Propriétés de la classe** et **Propriétés de la commandeX**, vous pouvez cliquer sur le bouton **Entrer le mot de passe** pour déverrouiller une bibliothèque protégée par mot de passe.
- La boîte de dialogue **Authentification** comprend deux options pour ignorer les invites de mot de passe que vous ne connaissez pas quand vous essayez d'accéder à plusieurs VIs protégés par mot de passe en une seule opération. Cliquez sur le bouton **Ignorer** pour ignorer une invite de mot de passe pour un VI ou cochez l'option **Ignorer les autres invites de mot de passe pour cette opération** et cliquez sur le bouton **Ignorer** pour ignorer tous les VIs protégés par mot de passe sans annuler l'opération.
- Sur la page **Taille de fenêtre** de la boîte de dialogue **Propriétés du VI**, le bouton **Définir à la taille de fenêtre actuelle** s'appelle désormais **Définir à la taille actuelle de la face-avant**.
- Sur la page **Tracés** de la boîte de dialogue **Propriétés du graphe**, l'option **Ne pas utiliser les noms de waveform pour les noms de tracés** est désormais l'option **Ignorer les attributs dynamiques ou waveform, y compris le nom des tracés**. Cette case à cocher n'est disponible que pour les graphes et graphes déroulants avec des données dynamiques ou waveforms.

Améliorations apportées aux palettes

Les améliorations suivantes ont été apportées aux palettes de LabVIEW 8.5 :

- Vous pouvez afficher ou masquer les catégories de palettes dans tous les formats d'affichage de la palette : **Catégorie (Standard)**, **Catégorie (Icônes et texte)**, **Icônes**, **Icônes et texte**, **Texte** et **Arbre**. Sur une palette fixée, cliquez sur le bouton **Affichage** de la barre d'outils de la palette et sélectionnez **Changer les catégories visibles** dans le menu local pour afficher toutes les catégories sur la palette **Commandes** ou **Fonctions**.

- La palette **VI et fonctions de contrôle d'applications** comprend la palette **Contrôle de mémoire**.

Autres améliorations apportées à l'environnement

Chaque fois que vous imprimez avec LabVIEW, l'imprimante que vous utilisez devient l'imprimante par défaut dans LabVIEW. LabVIEW ne reconnaît les changements d'imprimante par défaut du système que lorsque vous redémarrez LabVIEW.

Améliorations apportées aux projets LabVIEW

LabVIEW 8.5 comprend les améliorations suivantes apportées au projet LabVIEW et à ses fonctionnalités associées :

Améliorations apportées aux variables globales et à NI-PSP

Les améliorations suivantes ont été apportées aux variables partagées et à NI-PSP dans LabVIEW 8.5 :

Nœuds Variable partagée absolus ou relatifs à une cible

Vous pouvez définir un nœud Variable partagée comme absolu ou relatif d'après la façon dont vous voulez qu'il se connecte à la variable. Par défaut, lorsque vous créez une variable partagée à partir d'une cible dans un projet LabVIEW, les nœuds Variable partagée que vous créez à partir de la variable partagée sont absolus.



Remarque Vous pouvez créer, configurer et héberger des variables partagées uniquement sur les systèmes Windows. Vous pouvez utiliser les fonctions et VIs **DataSocket** pour lire ou écrire des variables partagées sur d'autres plates-formes. Si vous avez un VI qui possède un nœud Variable partagée configuré sur un système Windows, vous pouvez également déplacer ce VI sur une autre plate-forme.

Un nœud Variable partagée absolu se connecte à la variable partagée qui se trouve sur la cible sur laquelle vous avez créé la variable. Un nœud Variable partagée relatif à la cible se connecte à la variable partagée qui se trouve sur la cible sur laquelle vous exécutez le VI qui contient le nœud. Si vous déplacez vers une autre cible un VI qui contient un nœud Variable partagée relatif à la cible, vous devez aussi déplacer la variable partagée vers la nouvelle cible. Utilisez des nœuds Variable partagée relatifs à la cible si vous prévoyez de devoir déplacer des VIs et des variables vers d'autres cibles.

Pour changer la configuration du nœud Variable partagée d'absolu en relatif à la cible, faites un clic droit dessus à partir du diagramme et sélectionnez **Changer en relatif à la cible** dans le menu local. Pour changer la configuration du nœud de variable partagée de relatif à la cible à absolu, faites un clic droit dessus à partir du diagramme et sélectionnez **Changer en absolu** dans le menu local.

Implémentation TCP de NI-PSP (Windows, LabVIEW Real-Time)

Vous devez ouvrir des ports TCP en plus des ports UDP pour configurer des pare-feux et des routeurs NAT (Network Address Translating) afin de transmettre des variables partagées publiées sur réseau. En commençant au port TCP 59110, ouvrez l'un des ports TCP pour chaque application que vous effectuez. Par défaut, le protocole NI-PSP commence par rechercher les ports TCP disponibles, dans l'ordre croissant, à partir du port 59110 jusqu'à ce qu'il trouve un port disponible pour chaque application en cours d'exécution. Vous pouvez manuellement configurer la gamme des ports TCP que le protocole NI-PSP utilise en créant et en éditant un fichier `LogosXT.ini`. Pour les cibles fonctionnant sous LabVIEW Real-Time, vous pouvez configurer la gamme des ports TCP dans le fichier de configuration `ni-rt.ini`.

Résolution de conflits du projet

Le projet LabVIEW contient des conflits si plusieurs éléments du projet ont le même nom qualifié mais des chemins différents. LabVIEW effectue automatiquement le suivi de la hiérarchie de tous les éléments que vous intégrez au projet en balayant la hiérarchie du VI. Un triangle de mise en garde jaune apparaît sur tout élément en conflit. La fenêtre de l'**Explorateur de projet** offre plusieurs options pour résoudre des conflits.

- Cliquez sur le bouton **Résoudre les conflits** dans la fenêtre l'**Explorateur de projet** pour afficher la boîte de dialogue **Résoudre les conflits du projet**. Utilisez cette boîte de dialogue pour résoudre les conflits en renommant ou en redirigeant les éléments en conflit de façon à appeler des éléments dépendants à partir du chemin correct. Si le projet ne contient pas de conflit, LabVIEW désactive le bouton **Résoudre les conflits**.
- Dans la fenêtre de l'**Explorateur de projet**, sélectionnez **Projet» Afficher les chemins d'éléments** pour afficher la colonne **Chemins** fenêtre de l'**Explorateur de projet** et voir les chemins des fichiers qui correspondent aux éléments du projet. Vous pouvez aussi afficher la colonne **Chemins** en faisant un clic droit sur la racine du projet et en sélectionnant **Afficher» Chemins complets** dans le menu local.

- La fenêtre de l'**Explorateur de projet** comporte deux pages. La page **Éléments** affiche le contenu du projet. La page **Fichiers** affiche les éléments du projet qui ont chacun un fichier correspondant sur le disque. Vous pouvez organiser les noms de fichiers et les dossiers sur la page **Fichiers**, et effectuer des opérations sur disque (renommer, réorganiser ou supprimer, par exemple). Par exemple, si vous renommez un fichier sur la page **Fichiers**, LabVIEW renomme ce fichier sur le disque. Vous pouvez passer d'une page à l'autre en cliquant avec le bouton droit sur un dossier ou un élément situé sous une cible et en sélectionnant **Indiquer sur l'onglet Éléments** ou **Indiquer sur l'onglet Fichiers**. Les éléments de menu local **Indiquer sur l'onglet Éléments** et **Indiquer sur l'onglet Fichiers** apparaissent uniquement si l'élément existe dans l'autre onglet. Par exemple, les éléments du projet que vous n'avez pas enregistrés n'ont pas de fichier correspondant sur disque donc LabVIEW ne peut pas afficher l'élément dans l'onglet.
- Lorsque vous ouvrez un VI à partir d'un projet, LabVIEW ajoute tous les membres de la hiérarchie du VI qui ne se trouvent pas encore dans le projet dans **Dépendances**. LabVIEW organise les **Dépendances** en trois dossiers : `vi.lib`, `user.lib` et **Éléments** en mémoire. Lorsque vous ouvrez un VI que ne se trouve pas dans le projet, LabVIEW ajoute le VI dans **Éléments** en mémoire. Ces éléments restent dans le projet tant que le VI auquel ils font référence se trouve en mémoire. **Dépendances** est mis à jour chaque fois que vous ajoutez, supprimez ou enregistrez un élément dans le projet.

Utilisation de dossiers dans les projets LabVIEW

Vous pouvez ajouter deux types de dossiers à un projet LabVIEW, des dossiers virtuels et des dossiers remplis automatiquement. Les dossiers virtuels organisent les éléments du projet dans la fenêtre de l'**Explorateur de projet** et n'ont pas de valeur correspondante sur disque. Sur la page **Éléments**, cliquez avec le bouton droit sur une cible et sélectionnez **Ajouter»Dossier (Instantané)** dans le menu local pour ajouter un dossier virtuel au projet. Les dossiers remplis automatiquement sont automatiquement remplis et mis à jour pour refléter le contenu d'un dossier sur le disque. Sur la page **Éléments**, cliquez avec le bouton droit sur une cible ou un dossier et sélectionnez **Ajouter»Dossier (rempli automatiquement)** dans le menu local pour ajouter un dossier rempli automatiquement au projet. Le contenu des dossiers remplis automatiquement ne correspond pas toujours exactement au contenu du disque dans le cas des bibliothèques de projet. Le projet affiche le contenu des bibliothèques de projet (`.lvlib`) suivant la hiérarchie des bibliothèques et non pas suivant l'organisation sur disque. Par exemple, si un VI est membre d'une bibliothèque de projet, ce VI apparaît dans le

projet sous le fichier de bibliothèque. Le VI n'apparaît pas sous le dossier rempli automatiquement.

Nouvelles boîtes de dialogue de projet

LabVIEW 8.5 inclut les nouvelles boîtes de dialogue de projet LabVIEW suivantes :

- La boîte de dialogue **Résoudre le conflit de chargement** apparaît quand LabVIEW charge un fichier dont les dépendances sont en conflit avec d'autres éléments du projet et LabVIEW ne peut pas déterminer quel élément dépendant charger. Utilisez cette boîte de dialogue pour choisir quel fichier dépendant charger. Vous devez résoudre les conflits pour que le fichier puisse être chargé.
- La boîte de dialogue **Conflits de hiérarchie avec le projet** apparaît si vous essayez d'ouvrir un fichier qui est en conflit avec des éléments qui se trouvent déjà dans le projet LabVIEW. LabVIEW ne peut pas ouvrir le fichier car des éléments de la hiérarchie du VI ont le même nom qualifié que des éléments du projet ou de **Dépendances**. Utilisez cette boîte de dialogue pour résoudre des conflits entre éléments appartenant à deux hiérarchies de VIs du projet.
- La boîte de dialogue **Ajouter au projet et mettre les dépendances à jour** apparaît quand vous ouvrez le diagramme d'un VI de projet LabVIEW et que vous ajoutez un nouveau sous-VI au diagramme à partir du disque qui a le même nom qualifié qu'un élément déjà présent dans le projet. Utilisez cette boîte de dialogue pour ajouter un nouveau sous-VI au projet et mettre à jour des éléments appelants de façon à ce qu'ils fassent référence au chemin correct. Vous pouvez ajouter un élément au projet même s'il a des conflits avec d'autres éléments du projet. Vous devez résoudre les conflits pour que le fichier puisse être ouvert.
- La boîte de dialogue **Conflits de dépendance du projet détectés** apparaît quand vous essayez d'ajouter un fichier au projet qui peut causer des conflits dans le projet. Utilisez cette boîte de dialogue pour afficher les nouvelles dépendances en conflit que LabVIEW ajoute au projet. LabVIEW ajoute tous les membres de la hiérarchie du VI qui ne se trouvent pas déjà dans le projet dans **Dépendances**. Vous devez résoudre les conflits pour que le fichier puisse être ouvert.
- La boîte de dialogue **Annuler le changement de nom des fichiers** apparaît si vous essayez de renommer un fichier sur disque alors qu'il existe dans un dossier rempli automatiquement du projet LabVIEW. LabVIEW détecte le changement de nom sur disque et ne peut pas renommer l'élément et rediriger les éléments dépendants. Vous devez annuler le changement de nom sur disque et accepter le fichier en tant que nouveau fichier dans le projet puis résoudre les conflits susceptibles de se produire. Cette boîte de dialogue n'apparaît que si le

fichier que vous essayez de renommer est déjà ouvert dans LabVIEW ou qu'il a des appelants dans le projet. Utilisez cette boîte de dialogue pour annuler le changement de nom.

- La boîte de dialogue **Rechercher les appelants** apparaît si vous cliquez avec le bouton droit sur un élément de la fenêtre de l'**Explorateur de projet** et que vous sélectionnez **Rechercher» Appelants** dans le menu local. Utilisez cette boîte de dialogue pour trouver tous les appelants d'un élément spécifique du projet. Si l'élément n'a qu'un seul appelant, LabVIEW le met en évidence dans la fenêtre de l'**Explorateur de projet**.
- La boîte de dialogue **Rechercher les sous-VIs** apparaît si vous cliquez avec le bouton droit sur un élément de la fenêtre de l'**Explorateur de projet** et que vous sélectionnez **Rechercher»Sous-VIs** dans le menu local. Utilisez cette boîte de dialogue pour trouver tous les sous-VIs d'un élément spécifique du projet. Si l'élément n'a qu'un seul sous-VI, LabVIEW le met en évidence dans la fenêtre de l'**Explorateur de projet**.
- La boîte de dialogue **Rechercher les éléments sans appelants** apparaît si vous cliquez avec le bouton droit sur la racine du projet dans la fenêtre de l'**Explorateur de projet** et que vous sélectionnez **Rechercher les éléments sans appelants** dans le menu local. Utilisez cette boîte de dialogue pour trouver tous les éléments de niveau principal sans appelants du projet.
- La boîte de dialogue **Rechercher les conflits** apparaît si vous cliquez avec le bouton droit sur un élément en conflit dans la fenêtre de l'**Explorateur de projet** et que vous sélectionnez **Rechercher» Conflits** dans le menu local. Utilisez cette boîte de dialogue pour trouver tous les éléments du projet qui sont en conflit avec l'élément que vous sélectionnez. Si l'élément n'a qu'un seul conflit, LabVIEW le met en évidence dans la fenêtre de l'**Explorateur de projet**.

Autres améliorations apportées aux projets LabVIEW

Les autres améliorations suivantes ont été apportées aux projets dans LabVIEW 8.5 :

- Les fichiers `.lv1ps` enregistrent les paramètres de projet spécifiques à l'ordinateur local. Ne cochez pas les fichiers `.lv1ps` dans contrôle de source car les fichiers `.lv1ps` contiennent des paramètres spécifiques à l'ordinateur que vous utilisez. Par exemple, les fichiers `.lv1ps` contiennent la configuration locale du contrôle de source. LabVIEW enregistre le fichier `.lv1ps` quand vous enregistrez un projet et renomme ce fichier correctement quand vous renommez le projet. Vous pouvez supprimer des fichiers `.lv1ps` sans affecter les performances ou le comportement d'un projet car les fichiers `.lv1ps` contiennent seulement des paramètres de projet spécifiques à

l'ordinateur local. Si vous construisez une application, LabVIEW ne copie pas le fichier .lv1ps dans l'application construite.

- Recherchez des éléments dans un projet LabVIEW en utilisant la boîte de dialogue **Rechercher des éléments de projet**. Utilisez cette boîte de dialogue pour trouver des cibles, des VIs, des sous-VIs, des bibliothèques, des classes, et ainsi de suite. Sélectionnez **Édition» Rechercher des éléments de projet** dans la fenêtre de l'**Explorateur de projet** pour accéder à la boîte de dialogue **Rechercher des éléments de projet**. Cliquez avec le bouton droit sur un dossier, une bibliothèque de projet, une classe LabVIEW ou une commandeX dans un projet LabVIEW et sélectionnez **Rechercher des éléments de projet** dans le menu local pour rechercher des éléments de projet dans une bibliothèque ou un dossier spécifique. Vous pouvez aussi accéder à la boîte de dialogue **Rechercher les éléments de projet** à partir d'une fenêtre ou une fenêtre de classe de bibliothèque de projet autonome.
- Utilisez la fenêtre de l'**Explorateur de projet** pour ajouter un hyperlien comme élément dans le projet LabVIEW. Cliquez avec le bouton droit sur une cible ou sur un dossier ou une bibliothèque sous la cible et sélectionnez **Ajouter»Hyperlien** dans le menu local pour afficher la boîte de dialogue **Propriétés de l'hyperlien**. Vous pouvez utiliser les hyperliens pour créer des liens vers des fichiers ou des répertoires qui ne se trouvent pas sur l'ordinateur local mais qui sont accessibles via Internet ou votre réseau local. Les hyperliens peuvent correspondre aux adresses mailto, réseau, locales, HTTP, FTP, etc.
- Renommez une bibliothèque de projet LabVIEW, une commandeX ou une classe LabVIEW en cliquant avec le bouton droit sur cette bibliothèque de projet LabVIEW, commandeX ou classe LabVIEW et en sélectionnant **Renommer** dans le menu local pour entrer le nouveau nom.
- Le bouton **Tout enregistrer** sur la barre d'outils de la fenêtre de l'**Explorateur de projet** s'appelle désormais **Tout enregistrer (ce projet)**. Utilisez le bouton **Tout enregistrer (ce projet)** pour enregistrer tous les fichiers dans le projet actuel qui est ouvert.
- Sélectionnez **Fichier»Enregistrer sous** dans un projet LabVIEW préalablement enregistré pour afficher la nouvelle boîte de dialogue **Enregistrer sous**.

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

Les VIs et fonctions de LabVIEW 8.5 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 8.5 inclut les nouveaux VIs et fonctions suivants.

Fonctions et VIs de cluster, classe et variant

La palette **Cluster, classe et variant** comprend la nouvelle constante et le nouveau VI suivants.

- Obtenir la valeur par défaut de classe LabVIEW
- Objet LV

VIs de géométrie

La palette **Géométries** comporte le VI Créer du texte dans les systèmes de développement complet ou professionnel de LabVIEW.

VIs Mathématiques

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

- Ajustement de courbe non linéaire contraint
- Créer un polynôme à partir d'une décomposition en éléments simples
- Décomposition SVD généralisée
- Intervalles d'ajustement de courbe non linéaire
- Intégration numérique inégale
- Norme de vecteur

La nouvelle palette **Sous-programmes d'algèbre linéaire de base** comporte les nouveaux VIs suivants sur les systèmes de développement complet et professionnel :

- amax - Indice de l'élément max
- amin - Indice de l'élément min
- asum - Somme des valeurs absolues
- axpy - Produit Scalaire x Vecteur
- copy - Copie d'un vecteur
- ddot - Produit scalaire (DBL)
- dger - Mise à jour de rang 1 d'une matrice générale (DBL)
- drotm - Rotation de Givens rapide (DBL)
- drotmg - Paramètres de rotation de Givens rapide (DBL)
- dsymv - Produit Matrice symétrique x Vecteur (DBL)
- dsyr - Mise à jour de rang 1 d'une matrice symétrique (DBL)

- dsyr2 - Mise à jour de rang 2 d'une matrice symétrique (DBL)
- gemm - Produit Matrice x Matrice générales
- gemv - Produit Matrice générale x Vecteur
- nrm2 - Norme 2 d'un vecteur
- rot - Rotation de Givens
- rotg - Paramètres de rotation de Givens
- swap - Permutation d'un vecteur
- symm - Produit Matrice symétrique x Matrice
- syr2k - Mise à jour de rang 2k d'une matrice symétrique
- syrk - Mise à jour de rang k d'une matrice symétrique
- trmm - Produit Matrice triangulaire x Matrice
- trmv - Produit Matrice triangulaire x Vecteur
- trsm - Résoudre des équations linéaires (triangle, multiple)
- trsv - Résoudre des équations linéaires (triangle, unique)
- zdotc - Produit scalaire avec conjugaison (CDB)
- zdotu - Produit scalaire (CDB)
- zgerc - Mise à jour de rang 1 d'une matrice générale avec conjugaison (CDB)
- zgeru - Mise à jour de rang 1 d'une matrice générale (CDB)
- zhemm - Produit Matrice hermitienne x Matrice (CDB)
- zhmv - Produit Matrice hermitienne x Vecteur (CDB)
- zher - Mise à jour de rang 1 d'une matrice hermitienne (CDB)
- zher2 - Mise à jour de rang 2 d'une matrice hermitienne (CDB)
- zher2k - Mise à jour de rang 2k d'une matrice hermitienne (CDB)
- zherk - Mise à jour de rang k d'une matrice hermitienne (CDB)

Fonctions primitives VISA 64 bits

La palette **Accès aux registres** comprend les nouvelles fonctions suivantes :

- VISA In 64
- VISA Memory Allocation Ex
- VISA Move In 64
- VISA Move Out 64
- VISA Out 64

La palette **Accès de bas niveau aux registres** comprend les nouvelles fonctions suivantes :

- VISA Peek 64
- VISA Poke 64

Amélioration de l'allocation de mémoire et des performances des VIs avec la structure Élément en place

Pour de nombreuses opérations courantes, comme l'opération sur un élément d'un tableau et l'insertion de la valeur résultante au même indice du tableau, LabVIEW doit copier les valeurs des données et conserver ces valeurs en mémoire, ce qui augmente l'utilisation de mémoire.

Utilisez la structure Élément en place pour effectuer des opérations LabVIEW courantes au même emplacement de la mémoire sans que LabVIEW effectue plusieurs copies des valeurs des données en mémoire. La structure Élément en place utilise un ensemble de nœuds de bordure, qui sont des nœuds attachés à la bordure de la structure Élément en place, pour effectuer les opérations sur les données au même emplacement de la mémoire. Cliquez avec le bouton droit sur la bordure de la structure Élément en place et sélectionnez le nœud de bordure approprié pour les données.

Vous pouvez utiliser la structure Élément en place pour indexer un tableau, désassembler un cluster ou une waveform, effectuer des opérations sur des données variant ou sur n'importe quel type de données au même emplacement de la mémoire.

Reportez-vous à la rubrique *Structure Élément en place* du livre **Principes de base»Boucles et structures»Concepts** sous l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur la structure Élément en place.

Nœuds, fonctions et VIs modifiés

Les fonctions nœuds et VIs suivants ont été modifiés dans LabVIEW 8.5.

VIs et fonctions d'E/S sur fichiers

Les VIs suivants ont été modifiés dans la palette **E/S sur fichiers** :

- **Fermer les données de config** — Comprend une sortie **chemin du fichier** qui renvoie le chemin du fichier de configuration.
- **Fermer le stockage de données** — Comprend une sortie **chemin du fichier** qui renvoie le chemin de l'entrée **Refnum de stockage**.

VIs Mathématiques

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

- **Additionner des polynômes** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Additionner des polynômes rationnels** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Transformation inverse des vecteurs propres** — L'entrée **action** est maintenant recommandée et non plus requise.
- **Dérivée de $x(t)$** — Comprend une entrée **méthode** qui spécifie la méthode de différentiation.
- **Diviser des polynômes** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Diviser des polynômes rationnels** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Intégrale elliptique de 1er ordre** — Est un VI polymorphe comprenant les instances suivantes : Intégrale elliptique K complète et Intégrale elliptique F incomplète.
- **Intégrale elliptique de 2e ordre** — Est un VI polymorphe comprenant les instances suivantes : Intégrale elliptique E complète et Intégrale elliptique E incomplète.
- **Ajustement polynomial général** — Ce VI accepte désormais un **ordre polynomial** inférieur ou égal à 25. Si **ordre polynomial** est supérieur à 25, le VI définit les coefficients de **Coefficients du polynôme** à zéro si leur ordre est supérieur à 25 et renvoie une mise en garde.
- **Fonction Bêta (incomplète)** — Est un VI polymorphe comprenant les instances suivantes : Fonction Bêta et Fonction Bêta incomplète.
- **Fonction Gamma (incomplète)** — Est un VI polymorphe comprenant les instances suivantes : Fonction Gamma et Fonction Gamma incomplète.
- **Équations de Lyapunov** — Les deux instances de ce VI comprennent une entrée **type d'équation** qui spécifie le type d'équation Lyapunov.

- **Multiplier des polynômes** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Multiplier des polynômes rationnels** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Composition de polynômes** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Quadrature** — Est un VI polymorphe comprenant les instances suivantes : Quadrature 1D (VI) et Quadrature 1D (Formule).
- **Supprimer les coefficients nuls** — Les deux instances comprennent une entrée **type de seuil** qui spécifie la manière dont le VI supprime les éléments de fin.
- **Moyenne quadratique** — Ce VI est un VI polymorphe qui contient les instances suivantes : RMS (DBL) et RMS (CDB).
- **Soustraire des polynômes** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.
- **Soustraire des polynômes rationnels** — Les deux instances comprennent une entrée **seuil** qui spécifie le niveau auquel le VI supprime les éléments de fin et une entrée **type de seuil** qui spécifie de quelle manière le VI supprime les éléments de fin.

VIs de canaux de communication

(Linux) Les VIs suivants ont été modifiés dans la palette **Pipes** :

- **Close Pipe** — L'entrée **file descriptor** est désormais requise.
- **Open Pipe** — L'entrée **path to named pipe** est désormais requise.
- **Open System Command Pipe** — L'entrée **command line** est désormais requise.
- **Read From Pipe** — L'entrée **file descriptor** est désormais requise.
- **Write To Pipe** — Les entrées **file descriptor** et **data** sont désormais requises.

Fonctions et VIs de protocoles (Windows et Linux)

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

- **TCP Ouvrir une connexion** — Inclut une entrée port distant ou nom du service. Vous pouvez spécifier un nom de service pour que LabVIEW interroge NI Service Locator afin d'obtenir le numéro de port associé à ce nom de service.
- **TCP Créer un récepteur** — Inclut une entrée nom du service qui enregistre le nom du service et le numéro de port spécifié dans NI Service Locator.
- **TCP Créer un récepteur** — Inclut une entrée nom du service qui enregistre le nom du service et le numéro de port spécifié dans NI Service Locator.
- **UDP Ouvrir** — Inclut une entrée nom du service qui enregistre le nom du service et le numéro de port spécifié dans NI Service Locator.
- **UDP Écrire** — Inclut une entrée port ou nom du service. Vous pouvez spécifier un nom de service pour que LabVIEW interroge NI Service Locator afin d'obtenir le numéro de port associé à ce nom de service.

VIs de traitement de signaux

Les VIs suivants ont été modifiés dans la palette **Traitement du signal** :

- **Motif de rampe** — Comprend une entrée **type** qui spécifie le type de **Motif de rampe** à générer.
- **Mesures de transition** — Les deux instances comprennent les changements suivants :
 - La sortie **vitesse de variation** est devenue **pente**.
 - La sortie **durée** est devenue **durée de transition**.
 - La sortie **preshoot** est devenue **avant transition**. Cette sortie comprend un élément **undershoot** qui mesure la hauteur du minimum local qui précède une transition montante (ou descendante) et l'exprime en pourcentage de l'amplitude calculée sur la base de l'histogramme du signal. La sortie **avant transition** comprend aussi un élément **overshoot** qui mesure la hauteur du maximum local qui précède une transition montante (ou descendante) et l'exprime en pourcentage de l'amplitude calculée sur la base de l'histogramme du signal.
 - La sortie **overshoot** est devenue **après transition**. Cette sortie comprend un élément **undershoot** qui mesure la hauteur du minimum local qui suit une transition montante (ou descendante) et l'exprime en pourcentage de l'amplitude calculée sur la base de l'histogramme du signal. La sortie **après transition** comprend aussi un élément **overshoot** qui mesure la hauteur du maximum local qui suit une transition montante (ou descendante) et

l'exprime en pourcentage de l'amplitude calculée sur la base de l'histogramme du signal.

- **Dérrouler une phase** — Comprend une entrée **unité de phase** qui spécifie les unités de **Phase** et de **Phase déroulée**.
- **Propriétés de fenêtre** — Ce VI est un VI polymorphe avec les instances suivantes : Propriétés de fenêtre par coefficient et Propriétés de fenêtre par nom. L'instance Propriétés de fenêtre par nom de ce VI polymorphe comprend les entrées **taille**, **fenêtre** et **paramètre de la fenêtre**.

VIs et fonctions waveform

Les VIs suivants ont été modifiés dans la palette **Contrôle de waveform**.

Sur le VI Détection élémentaire du niveau de déclenchement, l'entrée **mode** a été renommée **mode d'emplacement**.

Palette Fonctions et VIs de cluster, classe et variant

La palette **Fonctions de cluster et variant** a été renommée **Fonctions et VIs de cluster, classe et variant**. La sous-palette **Fonctions d'attributs variant** a été renommée **Fonctions Variant**. Plusieurs objets ont été déplacés sur chaque palette.

La palette **Fonctions et VIs de cluster, classe et variant** comprend les nouvelles fonctions et les nouveaux VIs suivants :

- Tableau en cluster
- Construire un tableau de clusters
- Assembler
- Assembler par nom
- Appeler la méthode parente
- Cluster
- Cluster en tableau
- Obtenir la valeur par défaut de classe LabVIEW
- Indexer et assembler un tableau de clusters
- Objet LV
- En une classe plus générique
- En une classe plus spécifique
- Désassembler
- Désassembler par nom

La sous-palette **Fonctions Variant** inclut les fonctions suivantes :

- Supprimer un attribut de variant
- Chaîne aplatie en variant
- Obtenir un attribut de variant
- Définir un attribut de variant
- Fonction En variant
- Variant en donnée
- Variant en chaîne aplatie

Ajout d'un terminal de condition à une boucle For

Vous pouvez ajouter un terminal de condition pour configurer une boucle For pour qu'elle s'arrête prématurément si une condition booléenne ou une erreur se produit. Une boucle For avec un terminal de condition s'exécute jusqu'à ce qu'une condition booléenne ou une erreur se produise, ou jusqu'à ce que toutes les itérations soient terminées, suivant ce qui a lieu en premier. Pour ajouter un terminal de condition à une boucle For, cliquez avec le bouton droit sur le cadre de la boucle et sélectionnez **Terminal de condition** dans le menu local.

Reportez-vous au livre **Principes de base»Boucles et structures** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur les boucles For dans LabVIEW.

Utilisation de Nœuds de rétroaction pour enregistrer des données

Utilisez un nœud de rétroaction n'importe où sur le diagramme pour stocker les données d'une exécution précédente d'un VI ou d'une boucle. Lorsque vous placez un nœud de rétroaction sur le diagramme, LabVIEW y attache automatiquement un terminal d'initialisation.

Si vous placez un nœud de rétroaction dans une boucle, vous pouvez déplacer le terminal d'initialisation pour le mettre sur le bord gauche de la boucle soit en cliquant sur le terminal et en le faisant glisser soit en faisant un clic droit sur le nœud de rétroaction et en sélectionnant **Déplacer le terminal d'initialisation d'une boucle vers l'extérieur**. Si le terminal d'initialisation se trouve sur le bord gauche d'une boucle, vous pouvez sélectionner **Déplacer le terminal d'initialisation d'une boucle vers l'intérieur** ou **Initialiser globalement** dans le menu local pour le remettre sur le nœud. Si vous déplacez le terminal d'initialisation pour le mettre sur le bord gauche d'une boucle, le nœud de rétroaction s'initialise avant que la boucle dotée du terminal d'initialisation ne commence la première itération de chaque exécution. Si vous sélectionnez **Initialiser globalement** et que vous câblez une valeur en entrée au terminal

d'initialisation, le nœud de rétroaction s'initialise au premier appel du VI dans une exécution. Le terminal d'initialisation reste attaché au nœud et le nœud de rétroaction s'initialise globalement par défaut.

Si vous choisissez d'enregistrer un VI pour une version précédente de LabVIEW, LabVIEW remplace les nœuds de rétroaction à l'intérieur des boucles par des registres à décalage et remplace les nœuds de rétroaction à l'extérieur des boucles par des étiquettes.

Reportez-vous aux livres **Principes de base»Variables locales, variables globales et le nœud de rétroaction** et **Principes de base»Boucles et structures** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur les nœuds de rétroaction dans LabVIEW.

Autres modifications apportées aux VIs, fonctions et nœuds

D'autres modifications décrites ci-après ont été apportées aux fonctions, VIs et nœuds de LabVIEW 8.5 :

- Le VI Décompresser extrait le contenu d'un fichier zip qui n'est pas protégé par mot de passe dans le répertoire que vous spécifiez.
- Le VI Écrire dans un fichier son comprend désormais l'instance suivante : DBL Single. Pour les données de son mono, vous pouvez câbler un type de données waveform simple à l'entrée **données** de cette instance.
- **(Windows et Linux)** L'entrée **numéro de port** de la fonction Ouvrir une référence d'application s'appelle désormais **numéro de port ou nom du service**. Vous pouvez spécifier un nom de service pour que LabVIEW interroge NI Service Locator afin d'obtenir le numéro de port associé à ce nom de service.
- Les performances des VIs des palettes **Probabilités et statistiques**, **Fenêtres**, **Transformées** et **Génération de signaux** ont été améliorées.
- L'entrée **référence** et la sortie **référence de sortie** de la fonction Appeler une fonction d'une DLL sont devenues respectivement **chemin en entrée** et **chemin en sortie**.
- L'icône de la fonction Aperçu du premier élément d'une file d'attente a changé.
- Vous pouvez utiliser les propriétés NI_UpdateGroupName et NI_UpdateChannelName de la fonction TDMS - Définir des propriétés pour renommer des groupes et des voies dans des fichiers .tdms.

- Des icônes des VIs de la palette **Contrôle de périphérique d'entrée** ont été mis à jour.
- Les VIs Express Mesures spectrales acceptent désormais les types de données tableau de waveforms comme entrée.
- Le VI Express Formule accepte désormais comme entrée les types de données scalaire numérique, tableau 1D de numériques, tableau 2D de numériques, waveform et tableau de waveforms.
- **(Mac OS et Linux)** Les fonctions Copy et Move n'écrasent plus les fichiers ou les dossiers en lecture seule dans le chemin de destination, même si vous avez défini **overwrite** à TRUE.
- Il existe désormais de nouveaux noms et valeurs pour certaines **Constantes mathématiques et scientifiques** et **Constantes mathématiques et scientifiques Express**. Les valeurs de Constante de Planck, Charge élémentaire, Constante gravitationnelle, Constante d'Avogadro et Constante de Rydberg ont été mises à jour de façon à correspondre aux valeurs fournies par CODATA 2002.
- La fonction VISA Wait On Event et le VI Wait for RQS ne verrouillent plus d'autres opérations GPIB pendant leur exécution.
- La fonction Nombre en tableau de booléens accepte désormais les entiers 64 bits et le type de données en virgule fixe comme entrées.
- La méthode Classe LabVIEW : Créer de la classe Application inclut un paramètre nom que vous pouvez utiliser pour spécifier le nom de la classe LabVIEW.
- Les méthodes Recompiler et MassCompile des classes Application et Application (ActiveX) respectivement, incluent le paramètre Arrêté par utilisateur qui indique si l'utilisateur a arrêté l'opération de recompilation.
- Comme LabVIEW peut générer plusieurs événements Taille du panneau lorsque vous redimensionnez le panneau, LabVIEW génère désormais un événement final à la fin du redimensionnement dans lequel les paramètres **AncLmtes** et **NouvLmtes** renvoient la même valeur. Vous pouvez utiliser les valeurs pour identifier la fin du redimensionnement dans le code.
- La fonction Demander une désallocation se trouve désormais dans la palette **Contrôle de mémoire**.
- La fonction Fermer une référence accepte désormais le tableau 1D de type de données refnum comme entrée.
- Lors de l'utilisation de tableaux de nombres comme entrées du VI Express Formule, la longueur du tableau en sortie est désormais la même que le plus petit tableau en entrée.
- La fonction Adapter le type accepte des entiers 64 bits et des nombres à virgule flottante double-précision.

Nouveaux événements, propriétés, méthodes et classes

LabVIEW 8.5 comporte de nouveaux événements, propriétés, méthodes et classes de VI Serveur. Reportez-vous à la rubrique **Fonctionnalités et changements dans LabVIEW 8.5»Nouveaux objets de VI Serveur** de l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir une liste des nouveaux événements, méthodes, propriétés et classes.

Améliorations apportées à l'Application Builder

LabVIEW 8.5 comprend les améliorations suivantes de l'Application Builder.

Dans LabVIEW 8.5, la boîte de dialogue **Propriétés de la distribution du code source** contient la plupart des mêmes pages que les boîtes de dialogue **Application** et **Propriétés de la bibliothèque partagée**. Les boîtes de dialogue **Propriétés de l'application**, **Propriétés de la bibliothèque partagée** et **Propriétés de la distribution du code source** contiennent toutes les trois la plupart des mêmes changements. Les changements ci-dessous s'appliquent à ces trois boîtes de dialogue, sauf si indiqué autrement.

Page Informations

Les modifications suivantes ont été apportées à la page **Informations** dans LabVIEW 8.5 :

- Les zones de texte **Répertoire de destination de l'application**, **Répertoire de destination de la bibliothèque partagée** et **Répertoire de destination de la distribution** sont désormais appelées **Répertoire de destination**.
- Utilisez la zone de texte **Description de la spécification de construction** pour spécifier une description pour cette spécification.
- **(Distribution de code source)** Les options **Répertoire** et **Préserver la hiérarchie** de la page **Destinations** remplacent la fonctionnalité **Option d'envoi**.

Fichiers source

Les modifications suivantes ont été apportées à la page **Fichiers source** dans LabVIEW 8.5 :

- La liste **VI dynamiques et fichiers support** s'appelle désormais **Toujours inclus**.
- **(Distribution de code source)** Utilisez la liste **Toujours exclus** pour désigner les fichiers qui doivent être exclus de la construction.
- La boîte de dialogue **Propriétés de la distribution du code source** comprend la page **Fichiers source**. Vous devez indiquer quels fichiers

vous voulez inclure dans une distribution de code source. LabVIEW n'inclut plus automatiquement tous les fichiers dans le projet LabVIEW.

- Vous pouvez cliquer avec le bouton droit et sélectionner des fichiers dans l'arborescence **Fichiers du projet** pour ajouter plusieurs fichiers à la fois aux autres listes déroulantes.
- La liste **Vue de la destination** comprend les nouveaux emplacements suivants :
 - [DossierDonnéesAppCommun] — Les fichiers que vous placez dans ce dossier sont installés dans le dossier Application Data.
 - [DossierProgramFiles] — Les fichiers que vous placez dans ce dossier sont installés dans le dossier Fichiers communs.

Page Destinations

Les modifications suivantes ont été apportées à la page **Destinations** dans LabVIEW 8.5 :

- **La destination est une LLB** est devenue **LLB** dans la section **Type de destination**.
- Utilisez les options **Répertoire** et **Préserver la hiérarchie** pour indiquer que la destination de la construction est un répertoire et préciser si vous voulez préserver la hiérarchie de ce répertoire. Vous pouvez utiliser ces options avec des destinations personnalisées. De plus, ces options remplacent l'**Option d'envoi** de l'ancienne page **Paramètres de distribution** de la boîte de dialogue **Propriétés de la distribution du code source**.
- Utilisez l'option **Ajouter des fichiers à une nouvelle bibliothèque de projet** pour ajouter des fichiers à une nouvelle bibliothèque de projet créée lors du processus de construction. Désignez le nom de la bibliothèque de projet dans la zone de texte **Nom de la bibliothèque**.
- La boîte de dialogue **Propriétés de la distribution du code source** comprend la page **Destinations**.

Page Paramètres des fichiers source

Les modifications suivantes ont été apportées à la page **Paramètres des fichiers source** dans LabVIEW 8.5 :

- La section **Type d'inclusion** est un indicateur qui reflète la valeur que vous avez sélectionnée sur la page **Fichiers source**. Pour changer cette valeur, revenez à la page **Fichiers source** et déplacez l'élément sélectionné dans une autre liste déroulante.
- L'indicateur **Destination** est un menu déroulant. Vous pouvez sélectionner un autre répertoire de destination dans ce menu.

- La section **Paramètres du VI** a été déplacée et se trouve désormais dans la boîte de dialogue **Propriétés du VI**. Pour accéder à cette boîte de dialogue, cliquez sur le bouton **Personnaliser les propriétés du VI**. Les options de la boîte de dialogue **Propriétés du VI** correspondent aux options de la **Propriétés du VI** auxquelles vous accédez à partir du menu **Fichier**. Si vous modifiez des propriétés de VI dans cette boîte de dialogue, ces changements écrasent ceux que vous avez effectués dans la boîte de dialogue **Propriétés du VI** du menu **Fichier**.
- Utilisez les options de la section **Utiliser les paramètres d'enregistrement par défaut** pour supprimer le diagramme ou la boîte de dialogue de l'élément sélectionné.
- Utilisez les options **Aucune modification du mot de passe**, **Supprimer le mot de passe** et **Appliquer le nouveau mot de passe** pour configurer les paramètres du mot de passe pour le fichier sélectionné dans l'arborescence **Fichiers du projet**.
- Utilisez l'option **Renommer ce fichier dans la construction** pour renommer le fichier sélectionné dans l'arborescence **Fichiers du projet** durant le processus de construction. Si vous choisissez de renommer a fichier dans la construction, LabVIEW ajoute un préfixe aux éléments du dossier sélectionné.
- Si vous sélectionnez un dossier dans l'arborescence **Fichiers du projet**, vous pouvez choisir d'appliquer la plupart des options de la page **Paramètres des fichiers source** à tous les éléments du dossier.

Page Icône (Windows, Mac OS)

(Application) Utilisez le menu déroulant **Icon Image (Image de l'icône)** pour afficher les options de résolution et de couleur pour l'image d'une icône. L'aperçu de l'image affiche les options de résolution et de couleur que l'image supporte, jusqu'à une qualité couleur de 256 × 256 et 32 bits.

Page Avancé

Les modifications suivantes ont été apportées à la page **Avancé** dans LabVIEW 8.5 :

- L'option **Activer le support MathScript** n'existe plus. LabVIEW détecte automatiquement les appels MathScript à des fonctions définies par l'utilisateur, ou fichiers .m.
- Plusieurs options ont changé de place sur cette page afin de la rendre plus conviviale.

Page Autres exclusions

Les modifications suivantes ont été apportées à la page **Autres exclusions** dans LabVIEW 8.5 :

- **Déconnecter les définitions de type** — Déconnecte les définitions de type de la construction.
- **Supprimer les instances de VI polymorphe non utilisées** — Supprime toutes les instances de VI polymorphe non utilisées de la construction.
- **Supprimer les membres non utilisés des bibliothèques de projet** — Supprime les membres non utilisés de bibliothèques de projet de la construction.
 - **Modifier le fichier de bibliothèque de projet après avoir supprimé les membres non utilisés** — Modifie le fichier de bibliothèque de sorte qu'il ne fasse plus référence aux membres qui ont été supprimés.

Le tableau suivant affiche les options de LabVIEW 8.5 que vous pouvez sélectionner pour réaliser les mêmes exclusions qu'avec LabVIEW 8.2.

| Option de LabVIEW 8.2 | Option de LabVIEW 8.5 |
|--|--|
| Supprimer autant que possible | Sélectionnez ces trois nouvelles options. |
| Supprimer les membres sans référence des bibliothèques de projet | Sélectionnez Supprimer les membres non utilisés de bibliothèques de projet . Vous pouvez aussi sélectionner Modifier le fichier de bibliothèque de projet après avoir supprimé les membres non utilisés de façon à ce que le fichier de bibliothèque ne référence pas les membres supprimés. |
| Ne pas déconnecter les définitions de type et ne pas supprimer les membres non référencés | Ne sélectionnez aucune option. |

Page Informations sur la version (Windows, Mac OS)

Les informations sur la version des pages des boîtes de dialogue **Application** et **Propriétés de la bibliothèque partagée** s'affichent désormais sur la page **Informations sur la version**. Le nom de la spécification de construction remplit par défaut le champ **Description**. Les utilisateurs peuvent consulter cette information en cliquant avec le bouton droit sur l'application construite et en sélectionnant **Propriétés** dans le menu local.

Langues d'exécution

Vous pouvez sélectionner les langues supportées dans la liste **Langues supportées** en cochant la case située à côté de chaque langue.

Améliorations diverses apportées à l'Application Builder

- Dans la boîte de dialogue **Définir le prototype de VI**, le bouton radio **Conventions d'appel C** est activé par défaut.
- Cliquez avec le bouton droit sur **Spécifications de construction** dans la fenêtre de l'**Explorateur de projet** et sélectionnez **Tout construire** pour construire toutes les spécifications sous **Spécifications de construction**.
- Si vous convertissez un script de construction d'une version précédente de LabVIEW en une spécification de construction, LabVIEW place les éléments de la spécification de construction dans trois dossiers dans la fenêtre de l'**Explorateur de projet** au lieu de créer une hiérarchie.
- La boîte de dialogue **Propriétés du fichier zip** comprend la page **Structure de fichier zip**. Utilisez cette page pour spécifier la structure de fichier à utiliser pour construire le fichier zip et, éventuellement, renommer le répertoire de base dans lequel les fichiers sont décompressés.
- Vous pouvez inclure des sorties d'autres spécifications de construction comme des distributions de source, des bibliothèques partagées ou des applications dans le fichier zip que vous construisez en utilisant la boîte de dialogue **Propriétés du fichier zip**. Sur la page **Fichiers source**, sélectionnez la spécification de construction dans l'arborescence **Fichiers du projet** et utilisez le bouton fléché vers la droite pour l'ajouter à l'arborescence **Éléments à inclure**.

Les résultats des spécifications de construction de fichiers zip ou d'installateurs n'apparaissent pas dans l'arborescence **Fichiers du projet**. Pour inclure un autre fichier zip ou un installateur au fichier zip, ajoutez le fichier zip ou l'installateur que vous voulez inclure au projet LabVIEW sous la cible à partir de laquelle vous construisez le fichier zip.



Remarque Inclure les résultats d'autres spécifications de construction dans le fichier zip peut ralentir la génération de l'aperçu sur la page **Aperçu**.

Améliorations apportées à MathScript LabVIEW (Windows, pas dans la version de base)

Les améliorations et les modifications suivantes ont été apportées à MathScript dans LabVIEW 8.5 :



Remarque Pour afficher la **fenêtre MathScript LabVIEW**, sélectionnez **Outils»Fenêtre MathScript**.

Nouvelles fonctions MathScript

LabVIEW 8.5 inclut les nouvelles fonctions MathScript suivantes. Vous pouvez utiliser ces fonctions dans la **fenêtre MathScript LabVIEW** ou dans le nœud MathScript.

- classe `advanced` : `bessel`
- classe `approximation` : `interpft`
- classe `audio` : `soundsc` et `wavrecord`
- classe `basic` : `cumtrapz`, `feval`, `ipermute` et `permute`
- classe `dsp` : `cohere`, `csd`, `firpmord`, `psd` et `tfe`
- classe `geometry` : `cylinder`, `ellipsoid`, `inpolygon`, `rectint`, `sphere` et `voronoi`
- classe `integration` : `dblquad`, `quad8`, `quadl` et `triplequad`
- classe `linalg` : `eigs`, `funm`, `gsvd` et `linsolve`
- classe `membership` : `isscalar`
- classe `plots` : `axes`, `colormap`, `datetick`, `gca`, `get`, `ginput`, `gtext`, `image`, `pareto`, `rgbplot`, `set`, `surfnorm` et `view`
- classe `polynomials` : `pchip` et `polyder`
- classe `statistics` : `halton` et `richtmeyer`
- classe `string` : `evalc`, `regexp`, `regexpi`, `regexpexp`, `regexptranslate`, `strread` et `strtrim`
- classe `support` : `delete`, `dos`, `feof`, `fgetl`, `fgets`, `frewind`, `fseek`, `getfileproperty`, `imread`, `imwrite`, `lookfor`, `setfileproperty`, `system` et `textread`
- classe `vector` : `del2` et `divergence`
- classe `zerofinder` : `fsolve` et `fzero`

Améliorations du rapport d'erreur pour le nœud MathScript

Dans LabVIEW 8.5, le rapport d'erreur du nœud MathScript a été amélioré. Par exemple, si vous appelez une fonction définie par l'utilisateur, ou fichier `.m`, à partir d'un nœud MathScript, LabVIEW renvoie les erreurs de temps d'édition de la fonction définie par l'utilisateur à l'édition plutôt qu'à l'exécution. Si vous modifiez la fonction définie par l'utilisateur, LabVIEW met à jour la fenêtre **Liste des erreurs** dès que vous enregistrez le fichier `.m`. Cependant, si vous appelez une fonction à partir d'un nœud MathScript susceptible de changer la liste des chemins de recherche MathScript ou d'introduire de nouvelles variables à l'exécution, LabVIEW fonctionne avec une détection d'erreur réduite et les performances du nœud MathScript sont moins bonnes à l'exécution.

Un symbole de mise en garde apparaît sur le cadre du nœud MathScript pour indiquer que la détection d'erreur est réduite et que les performances sont moins bonnes à l'exécution. Les fonctions suivantes déclenchent l'apparition du symbole de mise en garde : `addpath`, `cd`, `clear`, `eval`, `evalc`, `load`, `path`, `rmpath` et `ui load`. La fonction `cd` et la fonction `path` ne causent l'apparition du symbole de mise en garde que si vous appelez ces fonctions avec une ou plusieurs entrées. Le symbole de mise en garde apparaît aussi si vous appelez une fonction définie par l'utilisateur qui appelle une de ces fonctions, si vous appelez un script défini par l'utilisateur ou si vous appelez une fonction définie par l'utilisateur qui appelle un script défini par l'utilisateur. Pour supprimer le symbole de mise en garde du nœud MathScript et améliorer les performances à l'exécution, effectuez les étapes suivantes.

1. Supprimez ces fonctions des scripts et des fonctions définies par l'utilisateur.
2. Supprimez les appels aux scripts définis par l'utilisateur. À la place, copiez le contenu du script défini par l'utilisateur dans le nœud MathScript ou la fonction définie par l'utilisateur qui appelle le script.
3. Ne modifiez pas la liste des chemins de recherche MathScript à l'exécution. Utilisez plutôt la page **Options MathScript : chemins de la recherche** pour configurer la liste de chemins de recherche par défaut.



Remarque Si un script d'un nœud MathScript appelle une fonction définie par l'utilisateur, LabVIEW utilise la liste des chemins de recherche par défaut pour lier l'appel de fonction au fichier `.m` spécifié. Après avoir configuré la liste des chemins de recherche par défaut et enregistré le VI qui contient le nœud MathScript, vous n'avez pas besoin de reconfigurer la liste des chemins de recherche de MathScript lorsque vous ouvrez le VI sur un autre ordinateur car LabVIEW recherche le fichier `.m` dans le répertoire où il se trouvait

la dernière fois que vous avez enregistré le VI. Vous devez toutefois maintenir le même chemin relatif entre le VI et le fichier .m.

Appel de fonctions définies par l'utilisateur à partir de MathScript LabVIEW

Lorsque vous appelez une fonction définie par l'utilisateur à partir de la **fenêtre MathScript LabVIEW**, LabVIEW effectue une recherche du haut vers le bas dans la liste des chemins de recherche MathScript pour trouver un fichier .m du nom spécifié. Si vous appelez une fonction définie par l'utilisateur à partir du nœud MathScript, LabVIEW recherche dans les trois emplacements suivants, dans l'ordre, un fichier .m du nom spécifié :

- En premier lieu, LabVIEW examine la mémoire pour déterminer si un fichier .m du nom spécifié se trouve déjà en mémoire.
- LabVIEW recherche ensuite dans le répertoire où se trouvait le fichier .m la dernière fois que vous avez enregistré le VI contenant le nœud MathScript. Comme LabVIEW cherche dans le répertoire où il a trouvé le fichier .m la dernière fois, il n'est pas nécessaire de reconfigurer la liste des chemins de recherche MathScript si vous ouvrez le VI sur un autre ordinateur. Vous devez toutefois maintenir le même chemin relatif entre le VI et le fichier .m.
- Pour finir, LabVIEW fait une recherche de haut en bas dans la liste des chemins MathScript. Si vous ouvrez le VI sur un ordinateur différent, la liste des chemins de la recherche risque de changer car LabVIEW recherche dans la liste de recherche MathScript configurée pour cet ordinateur, et non dans la liste des chemins de recherche configurée la dernière fois que vous avez enregistré le VI.



Remarque Si un symbole de mise en garde apparaît sur la bordure du nœud MathScript, LabVIEW ne regarde pas dans la mémoire ni dans le répertoire où le fichier .m se trouvait auparavant. Au lieu de cela, LabVIEW recherche un fichier .m du nom spécifié uniquement dans la liste des chemins de recherche MathScript.

Configuration du chemin de recherche MathScript

Utilisez la page **Options MathScript : chemins de la recherche** pour configurer le chemin de recherche par défaut et spécifier le répertoire de travail pour MathScript LabVIEW. Affichez cette page de l'un des emplacements suivants, suivant l'environnement pour lequel vous voulez configurer la liste des chemins de recherche :

- Pour configurer la liste des chemins de recherche de la **fenêtre MathScript LabVIEW**, affichez cette page à partir de la boîte de dialogue **Propriétés MathScript LabVIEW**.

- Pour configurer la liste des chemins de recherche de MathScript dans l'instance principale d'application, affichez cette page à partir de la boîte de dialogue **Options**.
- Pour configurer la liste des chemins de recherche pour les nœuds MathScript d'une cible d'un projet LabVIEW, affichez cette page à partir de la boîte de dialogue **Propriétés** de la cible.

Utilisation de la sonde MathScript LabVIEW

Utilisez la sonde MathScript LabVIEW pour visualiser les données d'un script d'un nœud MathScript lors de l'exécution d'un VI. Cliquez avec le bouton droit sur un nœud MathScript et sélectionnez **Sonde** dans le menu local pour utiliser la sonde MathScript.

La sonde MathScript affiche la liste de toutes les variables que vous définissez dans le script et donne un aperçu des variables que vous sélectionnez. La sonde MathScript affiche aussi le résultat généré à partir de script par MathScript. À la différence des sondes fournies, vous ne pouvez pas configurer la sonde MathScript pour répondre aux données. Par exemple, vous ne pouvez pas spécifier si vous voulez définir un point d'arrêt conditionnel.



Remarque Vous ne pouvez pas utiliser les modes animation de l'exécution, pas à pas et les points d'arrêt dans le script d'un nœud MathScript.

Définition des attributs de tracé dans MathScript LabVIEW

Vous pouvez obtenir et définir des attributs de tracé pour les types d'objets suivants dans les tracés de MathScript LabVIEW : les objets ligne, les objets zone de tracé, les objets fenêtre de tracé et les objets texte.

Pour créer et renvoyer un nouvel objet de ligne, utilisez les fonctions MathScript suivantes : `line`, `loglog`, `plot`, `semilogx`, ou `semilogy`. Pour créer un nouvel objet zone de tracé ou pour renvoyer un objet pour la zone de tracé existante, utilisez les fonctions MathScript suivantes : `axes`, `gca`, ou `subplot`. Pour créer un nouvel objet fenêtre de tracé ou pour renvoyer un objet pour la fenêtre de tracé existante, utilisez les fonctions MathScript suivantes : `clf`, `gcf`, ou `figure`. Pour créer et renvoyer un nouvel objet de texte, utilisez les fonctions MathScript suivantes : `text`, `title`, `xlabel`, ou `ylabel`.

Améliorations apportées aux boîtes de dialogue

Dans LabVIEW 8.5, les améliorations suivantes ont été apportées aux boîtes de dialogue MathScript.

Nouvelle boîte de dialogue MathScript

LabVIEW 8.5 intègre la boîte de dialogue **Propriétés de MathScript LabVIEW**. Pour afficher cette boîte de dialogue, sélectionnez **Fichier»Propriétés de MathScript LabVIEW** dans la **fenêtre MathScript LabVIEW**. Utilisez cette boîte de dialogue pour configurer les paramètres de la **fenêtre MathScript LabVIEW** et du moteur MathScript. Cette boîte de dialogue comprend les pages suivantes :

- **MathScript : options de la fenêtre** — Utilisez cette page pour configurer les paramètres, comme le style de police et le format d’affichage des nombres, de la **Fenêtre MathScript LabVIEW**.
- **Options MathScript : chemins de la recherche** — Utilisez cette page pour configurer la liste de chemins de recherche par défaut et spécifier le répertoire de travail pour MathScript LabVIEW.

Autres améliorations apportées aux boîtes de dialogue

Les boîtes de dialogue MathScript de LabVIEW 8.5 contiennent les autres améliorations suivantes :

- La boîte de dialogue **Préférences MathScript** n’existe plus. À la place, utilisez les pages **Options MathScript : chemins de la recherche** et **MathScript : options de fenêtre** de la boîte de dialogue **Propriétés de MathScript LabVIEW**.
- La boîte de dialogue **Préférences** n’existe plus. Utilisez la page **MathScript : options de fenêtre** de la boîte de dialogue **Propriétés de MathScript LabVIEW** pour configurer les paramètres qui se trouvaient sur les pages **Historique** et **Sortie**.
 - Les options de l’ancienne page **Infos système** n’existent plus.
 - L’option **Taille du buffer de l’historique** de la page **Historique** est devenue l’option **Buffer d’historique** et se trouve maintenant sur la page **MathScript : options de fenêtre** de la boîte de dialogue **Propriétés de MathScript LabVIEW**.
- L’**Éditeur de script** comporte un bouton **Nouveau script** qui efface le script qui se trouve dans l’**Éditeur de script**.

Améliorations apportées aux éléments de menu

Dans LabVIEW 8.5, les menus de la **fenêtre MathScript LabVIEW** ressemblent plus à ceux de LabVIEW. Les menus de la **Fenêtre LabVIEW MathScript** comprennent les améliorations suivantes :

- Le menu **Fichier** contient les changements suivants :
 - L'élément **Enregistrer le script** est devenu **Enregistrer**.
 - L'élément **Enregistrer le script sous** est devenu **Enregistrer sous**.
 - L'élément **Enregistrer et compiler le script sous** n'existe plus.
- Dans le menu **Édition**, l'élément **Supprimer** n'existe plus.
- Le menu **Affichage** contient les changements suivants :
 - L'élément **Espace de travail** n'existe plus. Vous ne pouvez pas masquer l'espace de travail de la **fenêtre MathScript LabVIEW**.
 - L'élément **Retour à la ligne pour le résultat** n'existe plus. Sélectionnez l'option **Retour à la ligne pour le résultat** sur la page **MathScript : options de fenêtre** de la boîte de dialogue **Propriétés de MathScript LabVIEW** pour activer le retour à la ligne dans la **Fenêtre de résultat**.
- Le menu **Exécution** comporte l'élément **Exécuter le script** qui exécute toutes les commandes qui se trouvent dans l'**Éditeur de script**.
- Les éléments **Charger des données**, **Enregistrer les données** et **Charger un script** sont passés du menu **Fichier** au menu **Exécution**.

Diverses améliorations et modifications apportées à MathScript

Les modifications supplémentaires suivantes ont été apportées à MathScript dans LabVIEW 8.5 :

- Les performances des scripts de LabVIEW MathScript qui contiennent des opérations d'indexation ont été optimisées afin de permettre une exécution plus rapide.
- Si vous appelez une fonction définie par l'utilisateur à partir d'un nœud MathScript qui ne contient pas de symbole de mise en garde, LabVIEW s'exécutera plus rapidement en raison des optimisations de performances dont bénéficie le Nœud MathScript.
- Lorsque vous référencez un fichier `.m` à partir d'un VI qui fait partie d'un projet LabVIEW, LabVIEW ajoute le fichier `.m` aux dépendances de la cible.

- Par défaut, le nœud MathScript affiche les numéros de ligne. Vous pouvez afficher et masquer les numéros de ligne dans le nœud MathScript, dans la boîte de calcul, dans le nœud de script MATLAB et dans le nœud de script Xmath. Pour afficher ou masquer les numéros de ligne, cliquez avec le bouton droit sur le nœud et sélectionnez **Éléments visibles»Numéros de ligne** dans le menu local.
- Vous pouvez redimensionner les différentes sections de la **fenêtre MathScript LabVIEW**, comme la **Fenêtre de commande** et la **Fenêtre de résultat**. Pour redimensionner une section, déplacez la barre de division qui sépare les deux sections que vous voulez redimensionner.
- Les fonctions suivantes comprennent une sortie **obj** qui renvoie la référence aux objets ligne, zone de tracé, fenêtre de tracé ou texte : `clf`, `figure`, `gcf`, `line`, `loglog`, `plot`, `semilogx`, `semilogy`, `subplot`, `text`, `title`, `xlabel` et `ylabel`.
- Les fonctions suivantes comprennent une entrée **nom** qui spécifie le nom d'attribut et une entrée **valeur** qui spécifie la valeur de l'attribut : `figure`, `line`, `loglog`, `plot`, `semilogx`, `semilogy`, `text`, `title`, `xlabel` et `ylabel`.
- La fonction `waitforbuttonpress` attend un événement Touche relâchée ou Souris relâchée avant de poursuivre l'exécution. Dans une fenêtre de tracé, l'événement Souris relâchée doit se produire dans la zone de tracé ou en dehors de la fenêtre de tracé. Si vous cliquez sur la barre de défilement, la barre de menus, la barre de titre ou la barre d'outils, cette fonction ne poursuit pas l'exécution.
- La fonction `legend` comprend une entrée **emplacement** qui spécifie à quel endroit du tracé la légende doit être ajoutée. Vous pouvez aussi utiliser la syntaxe `legend off` pour masquer la légende.
- Les fonctions `rand` et `randn` comprennent une entrée **s** qui spécifie la valeur de départ du générateur de nombres pseudo-aléatoires à utiliser au prochain appel de la fonction. Ces fonctions comprennent aussi une sortie **d** qui renvoie la valeur de départ du générateur de nombres pseudo-aléatoires.
- Les fonctions `butter`, `cheby1`, `cheby2` et `ellip` comprennent une entrée '**s**' qui indique à LabVIEW qu'un filtre analogique doit être conçu.
- L'entrée **fs** de la fonction `remezord` est désormais facultative et non plus obligatoire. La valeur par défaut est 2.
- L'entrée **b** de la `zplane` est désormais facultative et non plus obligatoire. Si **a** est un vecteur ligne et que vous ne spécifiez pas **b**, le pôle est (0, 0). Si **a** est un vecteur colonne ou une matrice et que vous ne spécifiez pas **b**, LabVIEW ne trace pas de pôle.

- La commande `help` supporte la syntaxe `help browser` qui donne une vue d'ensemble de la **fenêtre MathScript LabVIEW**.
- Vous pouvez inclure les fonctions suivantes dans une application autonome ou une bibliothèque partagée : `fopen`, `fclose`, `fread`, et `fwrite`.

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.

Programmation orientée objet LabVIEW

Les améliorations et modifications suivantes ont été apportées à la programmation orientée objet dans LabVIEW 8.5.

Accès aux données de classe LabVIEW

Vous pouvez utiliser la boîte de dialogue **Créer un VI d'accès** pour créer rapidement des VIs membres capables d'accéder aux données de classe LabVIEW. Vous pouvez créer des VIs d'accès dynamique ou statique pour écrire des données de classe ou les lire. Le VI d'accès en lecture désassemble les données de la classe LabVIEW pour vous permettre d'accéder aux données dans un VI appelant. Le VI d'accès en écriture assemble des nouvelles valeurs pour les données de la classe que vous transmettez par le VI appelant.

Pour vous faire gagner du temps de développement, vous pouvez sélectionner plusieurs éléments des données de la classe pour créer plusieurs VIs d'accès à la fois. Vous pouvez aussi gagner du temps en créant des VIs de lecture et d'écriture en même temps. Envisagez d'utiliser des VIs d'accès comme point de départ pour la création de VIs plus complexes.

Vous devez enregistrer la classe LabVIEW avant d'utiliser cette boîte de dialogue pour créer des VIs d'accès.

Vous disposez des méthodes suivantes pour lancer la boîte de dialogue **Créer un VI d'accès** à partir de la fenêtre de l'**Explorateur de projet** :

- Cliquez avec le bouton droit sur la classe LabVIEW et sélectionnez **Nouveau»VI d'accès aux membres de données** dans le menu local.
- Cliquez avec le bouton droit sur un membre de données dans la commande de données privées de la classe LabVIEW et sélectionnez **Créer un VI d'accès** dans le menu local.



Remarque Vous devez enregistrer une nouvelle classe LabVIEW avant d'utiliser l'une de ces options. LabVIEW grise les options **VI d'accès aux membres de données** et **Créer un VI d'accès** si vous n'avez pas enregistré la nouvelle classe.

Ouverture d'implémentations de sous-VIs de dispatch dynamique

Les sous-VIs de dispatch dynamique peuvent appeler n'importe quel ensemble de VIs d'une hiérarchie de classes LabVIEW. LabVIEW détermine quelle implémentation du sous-VI appeler à l'exécution en fonction du type de données de classe arrivant dans le terminal de dispatch dynamique.

Double-cliquez sur un sous-VI de dispatch dynamique pour afficher la boîte de dialogue **Choisir une implémentation**. Vous pouvez utiliser cette boîte de dialogue pour afficher toutes les implémentations d'un sous-VI de dispatch dynamique qui se trouvent actuellement en mémoire puis ouvrir une ou plusieurs implémentations du sous-VI.

Changements apportés aux menus locaux

Les changements suivants ont été apportés aux menus locaux des fonctionnalités de programmation orientée objet dans LabVIEW 8.5 :

- L'option de menu local **Nouveau»VI dynamique** est devenue **Nouveau»VI à partir d'un modèle de dispatch dynamique**. Vous pouvez accéder à cette option en cliquant avec le bouton droit sur une classe LabVIEW.
- Dans la version anglaise de LabVIEW, l'option de menu local **New»Override VI** est devenue **New»VI for Override**. Vous pouvez accéder à cette option en cliquant avec le bouton droit sur une classe LabVIEW.
- Cliquez avec le bouton droit sur une constante de classe, une commande ou un indicateur sur la face-avant ou le diagramme et sélectionnez **Afficher la bibliothèque de classe** dans le menu local pour mettre en évidence la classe dans la fenêtre de l'**Explorateur de projet**. Si la classe avec laquelle vous travaillez ne se trouve pas dans un projet LabVIEW, LabVIEW ouvre une fenêtre de classe pour afficher la classe.

Autres améliorations apportées à la programmation orientée objet LabVIEW

Les autres améliorations suivantes ont été apportées à la programmation orientée objet dans LabVIEW 8.5 :

- Utilisez des VIs membres de dispatch dynamique dans une classe LabVIEW pour tirer parti de la récursion. Les VIs récursifs peuvent s'appeler eux-mêmes sur leur propre diagramme, y compris sur le diagramme de sous-VIs. La récursion est utile si vous voulez effectuer de multiples opérations sur la sortie du même processus. Vous pouvez configurer un VI membre pour permettre la récursion en utilisant l'option **Partager les copies entre instances** sur la page **Propriétés de l'exécution** de la boîte de dialogue **Propriétés du VI**.
- Vous pouvez créer un VI membre à partir d'un modèle de dispatch statique en cliquant avec le bouton droit sur une classe LabVIEW et en sélectionnant **Nouveau»VI à partir d'un modèle de dispatch statique**. LabVIEW dote le VI membre de clusters d'**entrée d'erreur** et **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. Contrairement à ce qui se passe quand vous créez un VI de dispatch dynamique, LabVIEW ne définit pas les terminaux d'entrée et de sortie comme étant dynamiques sur le connecteur du VI de dispatch statique.
- Si vous créez une nouvelle classe LabVIEW, LabVIEW vous invite à la nommer lors de sa création.

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.

Changements des icônes de classe LabVIEW et de bibliothèque de projet

L'icône par défaut d'une bibliothèque de projet et des classes LabVIEW que vous créez dans LabVIEW 8.5 ne contient pas de nombre sur le masque d'icône et comporte des couleurs de masque différentes qui sont plus compatibles avec des icônes noir et blanc. Pour laisser la place nécessaire au nouveau masque et éviter que le masque d'icône de bibliothèque ou de classe ne cache une partie de l'icône du VI, utilisez une image qui ne dépasse pas 32 pixels de large sur 19 de haut.

Vous pouvez aussi éditer l'icône de classe ou de bibliothèque pour qu'elle utilise un autre masque sur la page **Paramètres généraux** des boîtes de dialogue **Propriétés de la bibliothèque de classe** ou **Propriétés de la classe**.

Vous pouvez créer plus facilement des icônes de VI pour les VIs de dispatch dynamique que vous créez en utilisant l'option **Nouveau»VI de redéfinition** du menu local. Le VI résultant garde l'icône du VI ancêtre et y superpose l'icône de classe enfant.

Meilleure utilisation de la mémoire virtuelle

(Windows) LabVIEW 8.5 reconnaît les adresses de grande taille. Sur les systèmes 64 bits, LabVIEW peut accéder par défaut à 4 Go de mémoire virtuelle. Sur les systèmes 32 bits, LabVIEW peut accéder par défaut à 2 Go de mémoire virtuelle. Vous pouvez permettre à LabVIEW d'accéder à 4 Go de mémoire virtuelle maximum sous un système d'exploitation 32 bits, en modifiant les paramètres de configuration de démarrage Windows. L'augmentation de mémoire virtuelle à laquelle LabVIEW a désormais accès réduit les risques d'erreurs générales LabVIEW liées à l'allocation de mémoire lorsque vous travaillez avec de gros volumes de données.

- **(Windows Vista x64)** Par défaut, LabVIEW peut accéder à 4 Go de mémoire virtuelle au maximum.
- **(Windows Vista)** Pour permettre à LabVIEW d'accéder à 3 ou 4 Go de mémoire virtuelle, ouvrez la fenêtre de ligne de commande en tant qu'administrateur et utilisez les commandes `bcdedit` pour ajouter une entrée dans la zone de mémoire BCD (Boot Configuration Data). Pour ouvrir la fenêtre de ligne de commande en tant qu'administrateur, naviguez vers la fenêtre dans le menu **Démarrer** de Windows, cliquez avec le bouton droit sur le nom du programme et sélectionnez **Exécuter en tant qu'administrateur**.
- **(Windows XP/2000)** Pour permettre à LabVIEW d'accéder à 3 Go de mémoire virtuelle, vous pouvez ajouter le tag `/3GB` au fichier `boot.ini` de Windows sur la ligne qui spécifie la version de Windows à démarrer. Si vous avez plus de 4 Go de RAM physique, vous pouvez utiliser le tag `/PAE` au lieu du tag `/3GB` pour permettre à LabVIEW d'accéder à 4 Go de mémoire virtuelle au maximum.

Reportez-vous à la rubrique *Utilisation de mémoire par les VIs* dans la rubrique **Principes de base»Gestion des performances et de la mémoire»Concepts**, sur l'onglet **Sommaire** de l'*Aide LabVIEW*, pour obtenir des informations plus détaillées sur l'utilisation de la mémoire virtuelle dans LabVIEW.

Fusion de VIs

LabVIEW fournit une application de fusion que vous pouvez utiliser pour fusionner des différences entre un VI d'origine et deux révisions de ce même VI. Utilisez la boîte de dialogue **Fusionner des VIs** pour fusionner des changements entre des VIs.

Sélectionnez **Outils»Fusionner des VIs** sur la face-avant ou le diagramme pour afficher la boîte de dialogue **Fusionner des VIs**. Spécifiez le VI d'origine dans le champ **VI de base**. Spécifiez les deux VIs à fusionner dans les champs **Leur VI** et **Votre VI**. Cliquez sur le bouton **Fusionner** pour fusionner les VIs sélectionnés.

Vous pouvez aussi configurer un gestionnaire de contrôle de source tiers pour qu'il utilise `LVMerge.exe` comme application de fusion par défaut. `LVMerge.exe` est un programme qui est l'équivalent de la boîte **Fusionner des VIs**.

Reportez-vous au livre **Principes de base»Directives de développement»Procédures»Fusion de VIs** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur la fusion de VIs.

Améliorations apportées à l'Assistant Importer une bibliothèque partagée

Utilisez l'Assistant **Importer une bibliothèque partagée** pour créer ou mettre à jour une bibliothèque de projet LabVIEW de VIs wrapper pour les fonctions d'un fichier `.dll` de Windows, d'un fichier `.so` de Linux ou d'un fichier `.framework` de Macintosh. L'Assistant **Importer une bibliothèque partagée** comprend les améliorations diverses suivantes :

- **Support des pointeurs** — Vous pouvez importer des fonctions qui contiennent des structures avec des éléments de pointeurs. Vous pouvez aussi spécifier s'il faut allouer de la mémoire pour les données d'un pointeur avant que LabVIEW n'appelle la fonction.
- **Génération automatique de commandes personnalisées** — Pour les fonctions qui contiennent des structures, l'Assistant **Importer une bibliothèque partagée** crée automatiquement des commandes personnalisées pour les éléments de la structure, notamment des pointeurs.
- **Gestion totale du projet** — Si vous sélectionnez l'option **Mettre à jour les VIs d'une bibliothèque partagée** sur la page **Spécifiez le mode Créer ou Mettre à jour**, l'assistant conserve les paramètres les plus récents pour chaque fonction individuelle dans la bibliothèque partagée. Par exemple, si vous avez une bibliothèque partagée qui contient trois fonctions, vous pouvez mettre à jour uniquement la seconde fonction. La prochaine fois que vous exécuterez l'assistant sur

ce fichier de bibliothèque partagée, les paramètres d'origine pour la première et la troisième fonction seront conservés et les nouveaux paramètres seront appliqués uniquement à la seconde.

- **Support de la fonction améliorée Appeler une fonction d'une DLL** — Les VIs wrapper utilisent les améliorations de Appeler une fonction d'une DLL de LabVIEW 8.2. Ces améliorations comprennent le support d'un pointeur sur les données du tableau et la taille minimale d'un tableau, ainsi qu'une fonctionnalité améliorée de rapport des erreurs.
- **Détails sur la fonction** — Vous pouvez sélectionner une fonction sur la page **Sélectionnez les fonctions à convertir** pour afficher des informations détaillées sur cette fonction. L'assistant ajoute aussi des informations sur la fonction à l'aide contextuelle sur le VI wrapper.
- **Rapport d'erreur amélioré** — Si l'assistant trouve des erreurs ou des mises en garde alors qu'il analyse le fichier d'en-tête, ces mises en garde et des solutions possibles s'affichent lorsque vous sélectionnez la fonction sur la page **Sélectionnez les fonctions à convertir**. Cette fonctionnalité remplace la page **Mises en garde** de LabVIEW 8.2.
- **Génération d'indicateurs pour les valeurs renvoyées** — Vous pouvez choisir de créer un indicateur pour les valeurs renvoyées dans un VI wrapper.
- **Résultats de l'analyse du fichier d'en-tête mis en cache** — Si vous exécutez l'assistant à plusieurs reprises sur la même bibliothèque partagée et que vous ne modifiez pas le fichier d'en-tête, l'assistant met en cache les résultats de l'analyse du fichier d'en-tête pour optimiser les performances.
- **Page Configurez les chemins à inclure et les définitions de préprocesseurs** — La page **Configurez les chemins à inclure et les symboles prédéfinis** a été renommée **Configurez les chemins à inclure et les définitions de préprocesseurs**.

Reportez-vous au livre **Principes de base»Appel de code provenant de langages de programmation textuels»Concepts»Importation de bibliothèques partagées** sous l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations complémentaires sur l'Assistant **Importer une bibliothèque partagée**.

Améliorations du contrôle d'image 3D

Vous pouvez ajouter des objets texte dans une scène 3D. Le texte apparaît quand vous générez la scène 3D. Utilisez le VI Créer un objet et le VI Créer du texte pour ajouter un objet texte à une scène 3D. Vous pouvez utiliser les propriétés et les méthodes de TexteScène pour configurer les attributs de l'objet texte, comme la couleur ou la taille des caractères, et pour accéder aux polices qui ne sont pas installées sur le système local.

La méthode Définir un attribut traçable remplace la méthode Définir la géométrie pour appliquer une géométrie ou du texte à un objet 3D. La méthode Définir la géométrie est supportée par LabVIEW 8.5 pour les VIs dont le dernier enregistrement s'est effectué dans LabVIEW 8.2, mais vous ne pouvez pas câbler un objet texte à cette méthode.

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

Utilisation de structures cadencées pour la programmation multicœur

Le multitraitement, ou programmation multicœur, fait référence à la présence de plusieurs processeurs dans un ordinateur qui sont chacun capables d'exécuter simultanément un thread différent. Une application multithread peut avoir plusieurs threads qui s'exécutent en même temps sur différents processeurs. LabVIEW 5.0 était la première version de LabVIEW à intégrer la programmation multicœur. LabVIEW 5.0 séparait le thread d'interface utilisateur des threads d'exécution. Les diagrammes s'exécutaient dans un ou plusieurs threads, et les faces-avant étaient mises à jour dans un autre. Le système d'exploitation s'exécutait en mode multitâche préemptif entre les threads en accordant du temps sur le processeur au thread d'exécution puis au thread d'interface utilisateur, et ainsi de suite.

La boucle cadencée et les structures Séquence cadencée intègrent une entrée **Processeur** qui vous permet de confier manuellement la gestion de l'exécution des structures aux UC disponibles. Vous pouvez configurer l'affectation du processeur en câblant une entrée à l'entrée **Processeur** du nœud d'entrée de la structure ou des étapes de la structure. (**Windows**) Vous pouvez aussi configurer des processeurs de façon à gérer des structures cadencées dans la section **Affectation du processeur** des boîtes de dialogue **Configurer la boucle cadencée**, **Configurer la boucle cadencée à étapes**, **Configurer le cadencement de l'étape suivante** et **Configurer l'itération suivante**.

Si vous chargez un VI avec une structure cadencée provenant d'une version antérieure de LabVIEW, l'entrée **Processeur** est définie à 0 par défaut, où 0 représente le premier processeur disponible dans le système car toutes les structures cadencées s'exécutent automatiquement sur le processeur 0 dans les versions précédentes de LabVIEW.

Type de données en virgule fixe

Le type de données en virgule fixe est un type de données numérique qui représente un ensemble de nombres rationnels en utilisant des chiffres binaires, ou bits. Contrairement au type de données en virgule flottante, qui permet au nombre total de bits utilisés par LabVIEW pour représenter les nombres de varier, vous pouvez configurer les nombres à virgule fixe pour toujours utiliser un nombre de bits spécifique. Le matériel et les cibles qui ne peuvent stocker et traiter les données qu'avec un nombre de bits limité ou fixe peuvent alors stocker et traiter les nombres. Vous pouvez spécifier la gamme et la précision des nombres à virgule fixe.

Utilisez la représentation en virgule fixe si vous n'avez pas besoin de la fonctionnalité dynamique de représentation en virgule flottante ou si vous voulez utiliser une cible qui ne supporte pas l'arithmétique en virgule flottante, comme les cibles FPGA.

Pour définir un nombre en représentation en virgule fixe, cliquez avec le bouton droit sur l'objet numérique et sélectionnez **Représentation** dans le menu local pour changer le type de données de l'objet. Vous pouvez aussi configurer l'encodage, la longueur de mot, la longueur de mot de la partie entière, la gamme et le delta désiré des nombres à virgule fixe et spécifier comment la fonction **Numérique** traite les conditions de dépassement et de quantification pour les nombres à virgule fixe. Pour configurer un nombre à virgule fixe, cliquez avec le bouton droit sur une constante, une commande, un indicateur ou une fonction numérique et sélectionnez **Propriétés** dans le menu local pour afficher la boîte de dialogue **Propriétés du numérique**, **Propriétés de la constante numérique** ou **Propriétés du nœud numérique**.

Reportez-vous à la rubrique *Données numériques* du livre **Principes de base»Construction du diagramme»Concepts** sur l'onglet **Sommaire** de l'*Aide LabVIEW* pour obtenir des informations plus détaillées sur le type de données en virgule fixe.

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.

Une fois que vous avez configuré LabVIEW pour qu'il utilise le contrôle de source, vous pouvez configurer les paramètres de contrôle de source sur des projets LabVIEW individuels. Configurer des paramètres de contrôle de source sur un projet LabVIEW individuel est utile si vous voulez utiliser un projet de contrôle de source différent de celui spécifié dans l'environnement LabVIEW ou si vous ne souhaitez pas utiliser de contrôle de source dans un projet LabVIEW particulier.

Les projets LabVIEW utilisent la configuration de contrôle de source si vous spécifiez l'environnement LabVIEW par défaut. Vous ne pouvez pas changer le gestionnaire de contrôle de source pour un projet LabVIEW. Les paramètres de contrôle de source pour des projets LabVIEW doivent correspondre à ceux du gestionnaire que vous spécifiez dans l'environnement LabVIEW. Vous ne pouvez utiliser qu'un seul gestionnaire de contrôle de source à la fois.

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. MATLAB® est une marque déposée de The MathWorks, Inc. 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. Pour consulter la liste des droits d'auteur, des conditions et des exclusions et limitations de garanties concernant les composants utilisés dans USI (Xerxes C++, ICU, HDF5, Citadel 5, b64 library et Stingray), veuillez vous référer à USICopyrights.chm.