



NATIONAL INSTRUMENTS™
LabVIEW™

Manuel de l'utilisateur

Filiales francophones

National Instruments France Centre d’Affaires Paris-Nord Immeuble “Le Continental” BP 217 93153 Le Blanc-Mesnil Cedex	National Instruments Suisse Sonnenbergstr. 53 CH-5408 Ennetbaden	National Instruments Belgium nv Ikaroslaan 13 B-1930 Zaventem	National Instruments Canada 1 Holiday Street East Tower, Suite 501 Point-Claire, Québec H9R 5N3
---	---	--	---

Support interne

E-mail :
france.support@ni.com
switzerland.support@ni.com
belgium.support@ni.com
canada.support@ni.com

Site FTP : ftp.ni.com

Adresse web :
ni.com
ni.com/france
ni.com/switzerland
ni.com/belgium
ni.com/canada

Support téléphonique en français

France	Tél. : 01 48 14 24 24	Fax : 01 48 14 24 14	
Suisse	Tél. : 056 200 51 51	Fax : 056 200 51 55	Tél. : 021 320 51 51 (Lausanne)
Belgique	Tél. : 02 757 00 20	Fax : 02 757 03 11	Tél. : 4050120 (Luxembourg)
Canada (Québec)	Tél. : 514 288 5722	Fax : 514 694 4399	

Les filiales

Afrique du Sud 11 805 8197, Allemagne 089 741 31 30, Australie 03 9879 5166, Autriche 0662 45 79 90 0, Brésil 011 3262 3599, Canada (Calgary) 403 274 9391, Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949, Canada (Toronto) 905 785 0085, Chine (Shanghai) 021 6555 7838, Chine (ShenZhen) 0755 3904939, Corée 02 3451 3400, Danemark 45 76 26 00, Espagne 91 640 0085, Finlande 09 725 725 11, Grande-Bretagne 01635 523545, Grèce 30 1 42 96 427, Hong Kong 2645 3186, Inde 91 80 535 5406, Israël 03 6393737, Italie 02 413091, Japon 03 5472 2970, Malaisie 603 9596711, Mexique 001 800 010 0793, Norvège 32 27 73 00, Nouvelle-Zélande 09 914 0488, Pays-Bas 0348 433466, Pologne 0 22 3390 150, Portugal 351 210 311 210, République tchèque 02 2423 5774, Russie 095 238 7139, Singapour 2265886, Slovénie 386 3 425 4200, Suède 08 587 895 00, Taiwan 02 2528 7227

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Pour plus d’informations de support, consultez l’annexe [Ressources techniques](#). Pour apporter des commentaires sur la documentation, envoyez un email à techpubs@ni.com.

Information importante

Garantie

Les supports sur lesquels vous recevez le logiciel National Instruments sont garantis comme ne tombant pas en panne pendant l'exécution des instructions de programmation, du fait des défauts de matériel et de fabrication, pendant une durée de 90 jours à compter de la date d'expédition prouvée par des reçus ou autres justificatifs. National Instruments, à sa convenance, réparera ou remplacera les supports du logiciel n'exécutant pas les instructions de programmation si National Instruments reçoit la notification de ces défauts pendant la période de garantie. National Instruments ne garantit pas que le fonctionnement du logiciel ne sera pas interrompu ou exempt d'erreurs.

Les équipements ne seront acceptés dans le cadre des travaux garantis que si un numéro d'Autorisation de Retour de Matériel (ARM) a été obtenu de l'usine et qu'il est clairement apposé à l'extérieur du paquet. National Instruments prendra à sa charge les frais de retour des pièces sous garantie.

National Instruments est convaincu que l'information contenue dans le présent document est exacte. Le présent document a été soigneusement revu afin d'en vérifier l'exactitude technique. En cas de présence d'erreurs techniques ou typographiques, National Instruments se réserve le droit de faire les modifications nécessaires sur les éditions ultérieures de ce document sans être tenu d'en informer préalablement les possesseurs de la présente édition. S'il suspecte des erreurs, le lecteur doit consulter National Instruments. National Instruments ne sera, en aucun cas, tenu responsable d'un quelconque dommage généré par ou en relation avec ce document ou l'information qu'il contient.

Hormis les dispositions prévues aux présentes, National Instruments ne donne aucune garantie, explicite ou implicite, et exclut spécifiquement toute garantie de commercialisation ou d'aptitude pour un usage particulier. Le droit du client à obtenir des dommages-intérêts en raison de l'erreur ou de la négligence de National Instruments sera limité au montant déboursé à ce titre par le client. National Instruments ne sera pas tenu de payer des dommages-intérêts en raison de la perte de données, de profits, d'utilisation des produits, ou des dommages-intérêts accessoires ou indirects, même s'il a été mis en garde contre la possibilité de tels dommages.

Cette limitation de responsabilité de National Instruments s'appliquera quelle que soit la forme de l'action, que son fondement soit contractuel ou quasi-délictuel, y compris la négligence. Toute action à l'encontre de National Instruments doit être intentée dans un délai d'un an après la survenance du fondement de ladite action. National Instruments ne sera pas tenue responsable du retard d'exécution pour des motifs échappant à sa volonté raisonnable. La garantie présentement accordée ne couvre pas les dommages, défauts, dysfonctionnements, ou défaillance de service provoqués par le non-respect par le propriétaire des instructions d'installation, utilisation, ou de maintenance de National Instruments ; de la modification du produit par le propriétaire ; de l'abus, de la mauvaise utilisation, ou des actes de négligence de la part du propriétaire ; ainsi que des pannes et des sautes d'électricité, de l'incendie, l'inondation, l'accident, les actions de tiers, ou d'autres événements échappant à sa volonté raisonnable.

Copyright

Conformément aux législations sur les droits d'auteur (« copyright »), il est interdit de reproduire ou de transmettre cette publication sous quelque forme que ce soit, électronique ou mécanique, y compris la photocopie, l'enregistrement, le stockage dans tout système d'introduction et d'accès, ou de la traduire, en tout ou partie, sans l'accord écrit préalable de National Instruments Corporation.

Marques déposées

Component Works™, DAQPad™, Data Socket™, HiQ™, HiQ-Script™, LabVIEW™, National Instruments™, NI™, NI-488™, NI-488.2™, ni.com™, NI-DAQ™, NI-FBUS™, NI-VISA™ et SCXI™ sont des marques déposées de National Instruments Corporation.

Les noms de produits et de sociétés cités sont des marques déposées par leurs propriétaires respectifs.

Brevets

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 et/ou à `ni.com/patents`.

AVERTISSEMENT CONCERNANT L'UTILISATION DES PRODUITS NATIONAL INSTRUMENTS

(1) Les produits National Instruments ne sont pas conçus avec des composants et un contrôle, destinés à satisfaire un niveau de fiabilité nécessaires à leur utilisation dans le cadre ou en relation avec des implants chirurgicaux, ou comme composants de première importance dans un système d'assistance à la vie quel qu'il soit et dont on peut raisonnablement penser que la défaillance de fonctionnement peut provoquer des blessures significatives à un être humain.

(2) Dans toute application, y compris celle qui précède, la fiabilité du fonctionnement des produits logiciels peut être altérée par des facteurs défavorables, comprenant de façon non limitative les fluctuations de l'alimentation en énergie électrique, les dysfonctionnements du matériel informatique, l'adéquation du logiciel du système d'exploitation de l'ordinateur, l'adéquation des compilateurs et du logiciel de développement utilisés pour développer une application, les erreurs d'installation, les problèmes de compatibilité entre le logiciel et le matériel, les dysfonctionnements ou les défaillances des systèmes de contrôle électronique, les défaillances passagères des systèmes électroniques, (matériel et/ou logiciel), les utilisations ou mauvaises utilisations non-prévues, ou erreurs de la part de l'utilisateur ou du développeur des applications (facteurs défavorables tels que ceux décrits collectivement ci-après sous la dénomination de "défaillances du système"). Toute application dont la défaillance du système provoquerait un risque de dommages aux biens ou aux personnes (y compris le risque de dommages corporels et de mort) ne devrait pas reposer uniquement sur un type de système électronique du fait du risque de défaillance du système. Pour éviter tout dommage, dommage corporel ou mort, l'utilisateur ou le développeur de l'application doit prendre des mesures de précaution raisonnables de protection contre les défaillances du système, comprenant de façon non limitative des mécanismes de sauvegarde ou d'interruption automatique. Chaque système, d'utilisateur final étant adapté aux besoins du client, diffère ainsi des plates-formes d'essais de National Instruments et parce qu'un utilisateur ou un développeur d'application peut utiliser des produits de National Instruments combinés avec d'autres produits de manière non-évaluée ou envisagée par National Instruments, l'utilisateur ou développeur de l'application est responsable, en dernier ressort, de la vérification et de la validation de l'adéquation des produits de National Instruments chaque fois que les produits de National Instruments sont incorporés dans un système ou application, y compris, sans limitation, la conception, le procédé et le niveau de sécurité appropriés dudit système ou application.

Sommaire

Avant-propos

Organisation de ce manuel.....	xx
Conventions	xx

PARTIE I Concepts LabVIEW

Chapitre 1

Introduction à LabVIEW

Ressources de documentation LabVIEW	1-1
VI's d'exemple et outils LabVIEW	1-4
VI's d'exemple LabVIEW	1-4
Outils LabVIEW.....	1-4

Chapitre 2

Introduction aux instruments virtuels

Face-avant.....	2-2
Diagramme.....	2-3
Terminaux.....	2-3
Nœuds.....	2-4
Fils de liaison.....	2-4
Structures.....	2-4
Icône et connecteur	2-5
Utilisation et personnalisation des VI's et sous-VI's	2-6

Chapitre 3

Environnement LabVIEW

Palette de commandes.....	3-1
Palette de fonctions	3-1
Parcourir les palettes de Commandes et de Fonctions.....	3-2
Palette d'outils	3-2
Les menus et la barre d'outils.....	3-3
Menus	3-3
Menus locaux.....	3-3
Menus locaux en mode Exécution	3-4
Barre d'outils.....	3-4

Personnalisation de votre environnement de travail.....	3-4
Personnalisation des palettes de commandes et de fonctions	3-4
Ajouter des VIs et des commandes à la bibliothèque utilisateur et à la bibliothèque d'instruments	3-5
Création et modification de l'apparence d'une palette.....	3-5
Comment LabVIEW enregistre les vues.....	3-6
Construction des sous-palettes ActiveX.....	3-6
Représentation des toolsets dans les palettes.....	3-6
Paramétrage des options de l'environnement de travail.....	3-6
Comment LabVIEW enregistre les options.....	3-7
Windows.....	3-7
Macintosh.....	3-7
UNIX.....	3-8

Chapitre 4

Construction de la face-avant

Configuration des objets de la face-avant	4-1
Afficher et masquer des éléments optionnels	4-2
Changement des commandes en indicateurs et vice versa.....	4-2
Remplacement des objets de la face-avant	4-2
Paramétrage des raccourcis clavier des commandes.....	4-3
Contrôler le comportement d'un bouton avec le raccourci clavier	4-3
Paramétrage de l'ordre de défilement des objets de la face-avant	4-4
Coloriage des objets	4-5
Utilisation d'images importées	4-5
Groupement et verrouillage des objets.....	4-5
Redimensionnement des objets	4-6
Mise à l'échelle des objets de la face-avant.....	4-6
Ajout d'espace dans la face-avant sans redimensionner la fenêtre.....	4-8
Commandes et indicateurs de la face-avant	4-8
Commandes et indicateurs 3D et classiques	4-8
Glissières, boutons rotatifs, cadrans et afficheurs numériques	4-9
Commandes et indicateurs à glissière(s)	4-9
Commandes et indicateurs rotatifs	4-9
Commandes et indicateurs numériques	4-10
Boîtes de couleur	4-10
Rampes de couleurs	4-10
Boutons, commutateurs et voyants	4-11
Affichages des zones de saisie de texte, étiquettes et chemins	4-11
Commandes et indicateurs de type chaîne.....	4-11
Commandes et indicateurs de type chemin	4-12
Chemins invalides	4-12
Chemins vides	4-12

Commandes et indicateurs de type tableau et cluster	4-12
Commandes à onglets	4-13
Listes déroulantes	4-13
Commandes et indicateurs de type énumération et menu déroulant	4-14
Commandes de type menu déroulant	4-15
Commandes de type énumération	4-15
Propriétés avancées des commandes et indicateurs de type énumération	4-16
Commandes et indicateurs de nom d'E/S.....	4-16
Commande waveform	4-17
Références à des objets ou à des applications	4-17
Commandes de boîte de dialogue.....	4-18
Étiquetage	4-18
Sous-titres	4-19
Caractéristiques du texte	4-19
Conception des interfaces utilisateur	4-21
Utilisation des commandes et indicateurs de la face-avant	4-21
Conception des boîtes de dialogue	4-22
Sélection de la taille de l'écran	4-22

Chapitre 5

Construction du diagramme

Relations entre les objets de la face-avant et les terminaux du diagramme.....	5-1
Objets du diagramme	5-1
Terminaux du diagramme.....	5-2
Type de données des indicateurs et des commandes	5-2
Constantes	5-4
Constantes universelles.....	5-5
Les constantes définies par l'utilisateur	5-5
Nœuds du diagramme	5-6
Sommaire des fonctions	5-7
Fonctions numériques.....	5-7
Fonctions booléennes	5-7
Fonctions chaîne.....	5-8
Fonctions tableau.....	5-8
Fonctions cluster.....	5-8
Fonctions de comparaison	5-9
Fonctions Temps & Dialogue.....	5-9
Fonctions E/S sur fichiers.....	5-9
Fonctions waveform	5-10
Fonctions de contrôle d'applications.....	5-10
Fonctions avancées	5-10
Ajout de terminaux aux fonctions du diagramme	5-11

Utilisation des fils de liaison pour relier les objets du diagramme.....	5-11
Câblage automatique des objets	5-13
Câblage manuel des objets	5-13
Sélection des fils de liaison	5-14
Suppression des fils de liaison brisés	5-14
Points de coercition	5-14
VIs et fonctions polymorphes.....	5-15
VIs polymorphes	5-15
Construction de VIs polymorphes	5-16
Fonctions polymorphes	5-18
Manipulation de données variant.....	5-18
Unités numériques et vérification de type stricte	5-19
Unités et vérification de type stricte	5-20
Flux des données dans le diagramme	5-22
Dépendance des données et dépendance des données artificielle.....	5-22
Dépendance de données manquante	5-23
Flux des données et gestion de mémoire	5-24
Conception du diagramme.....	5-25

Chapitre 6

Exécution et mise au point des VIs

Exécution des VIs.....	6-1
Configuration de l'exécution d'un VI	6-2
Correction des VIs brisés.....	6-2
Recherche des causes pour les VIs brisés	6-2
Causes courantes de VIs brisés	6-3
Techniques de mise au point	6-4
Mode Animation	6-4
Mode pas à pas	6-5
Outil Sonde	6-5
Points d'arrêt.....	6-6
Suspension de l'exécution	6-6
Détermination de l'instance actuelle d'un sous-VI.....	6-7
Mise “en commentaire” d’une section particulière du diagramme.....	6-7
Désactivation des outils de mise au point.....	6-8
Données indéfinies ou inattendues	6-8
Données par défaut et valeurs inattendues dans des boucles	6-9
Boucles For	6-9
Boucles While.....	6-9
Données par défaut dans les tableaux	6-9
Éviter des données indéfinies.....	6-10
Recherche et gestion d'erreur	6-10
Recherche d'erreur.....	6-10

Gestion d'erreur	6-11
Clusters d'erreur	6-12
Utilisation des boucles While pour la gestion d'erreur	6-12
Utilisation des structures Condition pour la gestion d'erreur	6-12

Chapitre 7

Création de VIs et de sous-VIs

Planification et conception de votre projet	7-1
Conception de projets avec plusieurs développeurs	7-2
Utilisation des VIs et fonctions intégrés	7-2
Construction de VIs et de fonctions de contrôle d'instruments et d'acquisition de données	7-3
Construction de VIs qui accèdent à d'autres VIs	7-3
Construction de VIs qui communiquent avec d'autres applications	7-4
Sous-VIs	7-4
Recherche d'opérations communes	7-5
Configuration du connecteur	7-6
Définition des entrées et sorties nécessaires, recommandées et optionnelles	7-8
Création d'une icône	7-8
Création de sous-VIs à partir des portions d'un VI	7-9
Conception de sous-VIs	7-9
Affichage de la hiérarchie des VIs	7-10
Enregistrement des VIs	7-10
Avantages de l'enregistrement des VIs en tant que fichiers individuels	7-11
Avantages de l'enregistrement des VIs sous des bibliothèques	7-11
Gestion des VIs dans les bibliothèques	7-12
Appellation des VIs	7-12
Enregistrement au format antérieur	7-12
Distribution des VIs	7-13
Construction d'applications autonomes et de bibliothèques partagées	7-14

PARTIE II

Construction et édition de VIs

Chapitre 8

Boucles For et While et Structures Condition, Séquence et Événement

Structures des boucles For et While	8-2
Boucles For	8-2
Boucles While	8-3
Éviter des boucles While infinies	8-4

Auto-indexation des boucles	8-4
Auto-indexation pour définir le décompte de la boucle For.....	8-5
Auto-indexation avec des boucles While	8-6
Registres à décalage sur des boucles.....	8-6
Contrôle de la synchronisation.....	8-7
Structures Condition, Séquence et Événement.....	8-7
Structures Condition	8-7
Valeurs du sélecteur de condition et types de données.....	8-8
Tunnels d'entrée et de sortie	8-9
Utilisation des structures Condition pour la gestion d'erreur	8-10
Structures Séquence	8-10
Éviter un usage excessif des structures Séquence	8-11
Programmation événementielle	8-13
Que sont les événements ?	8-13
Utilisation des événements avec LabVIEW.....	8-13
Structures Événement	8-14
Configuration des événements	8-15
Événements de type filtre ou notification.....	8-15
Exemple d'événement	8-17

Chapitre 9

Groupement des données au moyen de chaînes, de tableaux et de clusters

Chaînes	9-1
Chaînes sur la face-avant	9-2
Types d'affichage de chaîne.....	9-2
Tables	9-3
Édition de chaînes par programme.....	9-3
Formatage de chaînes	9-4
Spécificateurs de format	9-4
Numériques et chaînes	9-5
Conversion de types de données en XML	9-6
Utilisation de types de données basés sur le format XML	9-7
Schéma XML de LabVIEW	9-8
Groupement des données avec des tableaux et des clusters	9-9
Tableaux.....	9-9
Indices.....	9-9
Exemples de tableaux	9-9
Restrictions pour les tableaux	9-12
Création de commandes, d'indicateurs et de constantes tableau	9-12

Affichage de l'indice du tableau.....	9-13
Fonctions de tableaux.....	9-14
Redimensionnement automatique des fonctions de tableaux	9-14
Clusters	9-15

Chapitre 10

Variables globales et locales

Variables locales	10-1
Création de variables locales	10-2
Variables globales.....	10-2
Création de variables globales.....	10-3
Variables de lecture et d'écriture.....	10-3
Utilisation mesurée des variables globales et locales	10-4
Initialisation des variables globales et locales.....	10-5
Situations de compétition	10-5
Considérations sur la mémoire lors de l'utilisation de variables locales	10-6
Considérations sur la mémoire lors de l'utilisation de variables globales.....	10-6

Chapitre 11

Graphes et graphes déroulants

Types de graphes et de graphes déroulants.....	11-1
Options de graphe et de graphe déroulant	11-2
Échelles X et Y multiples sur des graphes et des graphes déroulants	11-2
Lissage pour les graphes et graphes déroulants.....	11-2
Personnalisation de l'apparence des graphes et des graphes déroulants.....	11-3
Personnalisation des graphes.....	11-4
Curseurs de graphe.....	11-4
Options d'échelle.....	11-5
Légende de l'échelle d'un graphe	11-5
Formatage des échelles d'un graphe.....	11-5
Utilisation du rafraîchissement progressif	11-6
Personnalisation des graphes déroulants	11-6
Longueur de l'historique.....	11-6
Mode de mise à jour du graphe déroulant.....	11-7
Courbes superposées ou courbes empilées	11-7
Graphes et graphes XY	11-8
Types de données des graphes à courbe unique	11-8
Graphes multi-courbes.....	11-9
Types de données des graphes XY à courbe unique	11-10
Types de données des graphes XY multi-courbes.....	11-11
Graphes déroulants	11-11

Graphes et graphes déroulants d'intensité	11-12
Représentation des couleurs.....	11-13
Options du graphe déroulant d'intensité.....	11-14
Options du graphe d'intensité.....	11-15
Graphes numériques	11-15
Masquage des données.....	11-17
Graphes 3D.....	11-17
Type de données waveform.....	11-19

Chapitre 12

VIs Graphisme et son

Utilisation de l'indicateur image.....	12-1
VIs de tracés d'images.....	12-3
Utilisation du VI Polar Plot comme sous-VI	12-3
Utilisation du VI Plot Waveform comme sous-VI.....	12-3
Utilisation des VIs Smith Plot comme sous-VIs.....	12-4
VIs Fonctions sur les images.....	12-5
Création et modification des couleurs avec les VIs Fonctions sur les images....	12-6
VIs Formats graphiques.....	12-7
VIs Son	12-8

Chapitre 13

E/S sur fichiers

Opérations de base d'E/S sur fichiers	13-1
Choix du format d'E/S sur fichiers	13-2
Quand utiliser des fichiers texte.....	13-2
Quand utiliser des fichiers binaires.....	13-3
Quand utiliser des fichiers journaux	13-4
Utilisation de VIs d'E/S sur fichiers de haut niveau.....	13-6
Utilisation de VIs et de fonctions d'E/S sur fichiers de bas niveau et avancés	13-7
Enregistrement sur disque en continu	13-8
Création de fichiers texte et de fichiers tableur.....	13-9
Formatage et écriture de données dans des fichiers	13-10
Balayage de données à partir de fichiers.....	13-10
Création de fichiers binaires.....	13-11
Création de fichiers journaux	13-11
Écrire des données waveform dans un fichier.....	13-12
Lecture de données waveform dans des fichiers	13-13
Paramètres dupliqués.....	13-14
Création de fichiers de configuration	13-14
Utilisation de fichiers de paramètres de configuration	13-15
Format de fichier de paramètres de configuration Windows	13-15

Enregistrement des données de la face-avant	13-17
Journaux de la face-avant automatiques et interactifs	13-17
Visualisation interactive des données enregistrées de la face-avant	13-18
Effacer un enregistrement	13-19
Effacer la liaison vers le fichier journal	13-19
Changer la liaison vers le fichier journal	13-19
Récupération des données de la face-avant par programmation	13-20
Récupération des données de la face-avant au moyen d'un sous-VI... 13-20	
Spécifier les enregistrements	13-21
Récupération des données de la face-avant au moyen des	
fonctions d'E/S sur fichiers.....	13-22

Chapitre 14

Documenter et imprimer des VIs

Documentation des VIs.....	14-1
Création de descriptions de VIs et d'objets.....	14-2
Activation de l'historique des révisions des VIs.....	14-2
Numéros de révision	14-2
Impression de la documentation.....	14-3
Enregistrement sous forme de fichiers HTML ou RTF	14-3
Sélection des formats d'image pour les fichiers HTML.....	14-4
Conventions d'appellation des fichiers image	14-5
Création de vos propres fichiers d'aide	14-5
Impression des VIs.....	14-5
Imprimer la fenêtre active	14-6
Impression de rapports	14-6
Impression par programmation.....	14-7
Impression à la fin de l'exécution.....	14-7
Utilisation d'un sous-VI pour effectuer une impression	
sélective à la fin de l'exécution.....	14-8
Techniques d'impression supplémentaires	14-8

Chapitre 15

Personnalisation des VIs

Configuration de l'apparence et du comportement des VIs	15-1
Personnalisation des menus	15-2
Création de menus	15-2
Gestion de la sélection des menus	15-3

Chapitre 16

Contrôle des VIs par programmation

Caractéristiques du VI Serveur.....	16-1
Construire des applications du VI Serveur.....	16-2
Références d'application et du VI.....	16-3
Édition des paramètres de l'application et du VI.....	16-4
Nœuds de propriété.....	16-4
Nœuds de propriété liés implicitement.....	16-4
Nœuds de méthode.....	16-5
Manipulation des propriétés et des méthodes de classe d'application.....	16-5
Manipulation des propriétés et des méthodes de classe de VI.....	16-6
Manipulation des propriétés et des méthodes de classe d'application et de VI... ..	16-7
Charger et appeler les VIs de façon dynamique.....	16-7
Nœuds d'appel par référence et refnums de VI de type strict.....	16-8
Édition et exécution de VIs sur des ordinateurs à distance.....	16-9
Contrôle des objets de la face-avant.....	16-9
Refnums de commande de type strict et refnums de commande partiellement spécifiés.....	16-9

Chapitre 17

Mise en réseau dans LabVIEW

Choix entre E/S sur fichiers, VI Serveur, ActiveX et mise en réseau.....	17-1
LabVIEW comme client et serveur réseau.....	17-2
Utilisation de la technologie DataSocket.....	17-2
Spécification d'une URL.....	17-3
Formats de données supportés par DataSocket.....	17-5
Utilisation de DataSocket sur la face-avant.....	17-5
Lecture et écriture de données en direct par l'intermédiaire du diagramme.....	17-7
DataSocket et données variant.....	17-8
Publication de VIs sur le Web.....	17-9
Options du serveur Web.....	17-9
Création de documents HTML.....	17-10
Publication d'images de faces-avant.....	17-10
Formats des images de faces-avant.....	17-10
Visualisation et contrôle des faces-avant à distance.....	17-11
Configuration du serveur pour les clients.....	17-11
Licence pour face-avant distante.....	17-12
Visualisation et contrôle de faces-avant dans LabVIEW ou à partir d'un navigateur Web.....	17-12
Visualisation et contrôle des faces-avant dans LabVIEW.....	17-12
Visualisation et contrôle de faces-avant à partir d'un navigateur Web.....	17-13

Fonctionnalité non supportée dans la visualisation et le contrôle des faces-avant distantes	17-14
Applications de communication de bas niveau.....	17-15
TCP et UDP.....	17-15
DDE (Windows).....	17-16
Apple Events et PPC Toolbox (Macintosh)	17-16
VIs de canaux (UNIX).....	17-16
Exécution des commandes système (Windows et UNIX).....	17-17

Chapitre 18

ActiveX

Objets, propriétés, méthodes et événements ActiveX	18-1
VIs, fonctions, commandes et indicateurs ActiveX	18-2
LabVIEW en tant que client ActiveX.....	18-3
Accès à une application pour laquelle ActiveX est activé	18-3
Insertion d'un objet ActiveX sur la face-avant.....	18-4
Définition des propriétés ActiveX.....	18-5
Navigateur de propriétés ActiveX.....	18-5
Pages de propriétés ActiveX	18-5
Nœuds de propriété	18-5
LabVIEW en tant que serveur ActiveX.....	18-6
Support pour les interfaces ActiveX Automation personnalisées	18-7
Utilisation de constantes pour définir des paramètres dans les VIs ActiveX	18-7

Chapitre 19

Appel de code provenant de langages de programmation textuels

Appeler une fonction d'une DLL	19-1
Code Interface Node	19-1

Chapitre 20

Formules et équations

Méthodes d'utilisation d'équations dans LabVIEW	20-1
Boîtes de calcul	20-2
Utilisation de la boîte de calcul	20-2
Variables dans la boîte de calcul	20-3
Nœuds d'expression	20-4
Polymorphisme dans les nœuds d'expression.....	20-5
Utilisation de HiQ avec LabVIEW	20-5
Nœuds de script HiQ et MATLAB.....	20-5
Suggestions de programmation pour les scripts HiQ et MATLAB	20-7
Fichiers support HiQ requis avec une application LabVIEW	20-8

Annexe A

Organisation de LabVIEW

Organisation de la structure des répertoires de LabVIEW	A-1
Bibliothèques	A-1
Structure et support	A-2
Exercices et exemples	A-2
Documentation	A-2
Fichier divers.....	A-2
Macintosh.....	A-3
Emplacement suggéré pour enregistrer les fichiers.....	A-3

Annexe B

Fonctions polymorphes

Conversion numérique.....	B-1
Polymorphisme pour les fonctions numériques	B-2
Polymorphisme pour les fonctions booléennes	B-4
Polymorphisme pour les fonctions de tableaux.....	B-5
Polymorphisme pour les fonctions de chaîne.....	B-5
Polymorphisme pour les fonctions de conversion de chaînes.....	B-6
Polymorphisme pour les fonctions Chaîne en nombre supplémentaires	B-6
Polymorphisme pour les fonctions de cluster.....	B-6
Polymorphisme pour les fonctions de comparaison.....	B-6
Polymorphisme pour les fonctions journal.....	B-8

Annexe C

Fonctions de comparaison

Comparaison de valeurs booléennes	C-1
Comparaison de chaînes.....	C-1
Comparaison de valeurs numériques.....	C-2
Comparaison des tableaux et clusters.....	C-2
Tableaux.....	C-2
Mode Comparer des éléments	C-2
Mode Comparer des ensembles.....	C-3
Clusters.....	C-3
Mode Comparer des éléments	C-3
Mode Comparer des ensembles.....	C-4

Annexe D

Masquage des données numériques

Annexe E
Ressources techniques

Glossaire

Index

Avant-propos

Ce manuel décrit l'environnement et les techniques de programmation graphique LabVIEW pour la construction des applications dans LabVIEW, telles que les applications de test et de mesure, d'acquisition de données, de commande d'instrument, de fichier journal, d'analyse de mesure et de génération de rapports.

Utilisez ce manuel pour apprendre à utiliser les fonctionnalités de programmation LabVIEW, y compris l'interface utilisateur LabVIEW et l'environnement de programmation, ainsi que les palettes, outils et boîtes de dialogue LabVIEW. Ce manuel n'inclut pas les informations spécifiques sur chaque palette, outil, menu, boîte de dialogue, commande ou encore VI et fonction intégrée. Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur ces éléments et pour recevoir des instructions pas à pas et détaillées sur l'utilisation des fonctionnalités LabVIEW et la construction d'applications spécifiques. Reportez-vous à la section *Ressources de documentation LabVIEW* du chapitre 1, *Introduction à LabVIEW*, pour de plus amples informations sur l'*Aide LabVIEW* et sur la façon d'y accéder.

Le *Manuel de l'utilisateur LabVIEW* est également disponible en version PDF. Si vous sélectionnez l'option d'installation **Complète**, LabVIEW installe les versions PDF de tous les manuels LabVIEW auxquels vous pouvez accéder en sélectionnant **Aide»Rechercher dans les manuels PDF...** dans LabVIEW.



Note Vous devez également avoir installé Adobe Acrobat Reader, version 4.0 ou ultérieure, pour lire les fichiers PDF. Reportez-vous au site Web d'Adobe Systems Incorporated à l'adresse www.adobe.com pour télécharger Acrobat Reader.

Vous pouvez accéder aux fichiers PDF à partir de l'*Aide LabVIEW*, mais pour cela, vous devez d'abord installer les fichiers PDF. Reportez-vous aux *Notes d'information* de LabVIEW ou aux *Notes de mise à jour* de LabVIEW pour de plus amples informations sur l'installation des versions PDF des manuels LabVIEW. Reportez-vous à la section *Ressources de documentation LabVIEW* du chapitre 1, *Introduction à LabVIEW*, pour de plus amples informations sur l'accès aux fichiers PDF dans la *Bibliothèque LabVIEW*.

Organisation de ce manuel

Le *Manuel de l'utilisateur LabVIEW* inclut deux sections. Partie I, *Concepts LabVIEW*, décrit les concepts de programmation pour la construction d'applications dans LabVIEW. Les chapitres de cette section vous présentent l'environnement de programmation LabVIEW et vous aident à planifier votre application.

Partie II, *Construction et édition de VIs*, décrit les fonctionnalités de LabVIEW, les VIs et les fonctions que vous pouvez utiliser pour faire fonctionner vos applications d'une certaine façon. Les chapitres de cette section décrivent l'utilité de chaque fonctionnalité LabVIEW et donnent une description de chaque classe de VIs et de leurs fonctions.

Conventions

Les conventions suivantes figurent dans ce manuel :

»

Le symbole » vous dirige vers les éléments de menu emboîtés et les options de boîtes de dialogue donnant lieu à une action finale. La séquence **Fichier»Mise en page»Options** vous indique que vous devez dérouler le menu **Fichier**, sélectionner l'élément **Mise en page** et sélectionner **Options** à partir de la dernière boîte de dialogue.



Cette icône signale une astuce qui vous donne des recommandations.



Cette icône signale une remarque qui vous donne des informations importantes.



Cette icône signale un avertissement qui vous indique les précautions à prendre pour éviter des dégradations, des pertes de données ou un blocage du système.

gras

Le texte en gras signale les éléments que vous devez sélectionner ou sur lesquels vous devez cliquer dans le logiciel, par exemple les éléments de menus et les options de boîtes de dialogue. Le texte en gras signale également les noms de paramètres, les commandes et boutons situés sur la face-avant, les boîtes de dialogue, les sections des boîtes de dialogue, les noms de menus et les noms de palettes.

italique

Le texte en italique signale les variables, la mise en valeur, une référence croisée ou une introduction à un concept clé. Cette police signale également le texte qui sert d'emplacement à un mot ou une valeur que vous devez fournir.

<code>monospace</code>	Cette police signale du texte ou des caractères que vous devez saisir à partir du clavier, des sections de code, des exemples de programmation et de syntaxe. Cette police est également utilisée pour les noms propres des unités de disque, chemins, répertoires, programmes, sous-programmes, sous-routines, pour les noms de périphériques, fonctions, opérations, variables, les noms et extensions de fichiers et les extraits de codes.
gras monospace	Le texte en gras de cette police signale les messages et réponses que l'ordinateur affiche automatiquement à l'écran. Cette police met aussi en relief les lignes de codes qui diffèrent des autres exemples.
<i>italique monospace</i>	Le texte en italique de cette police signale le texte qui sert d'emplacement à un mot ou une valeur que vous devez fournir.
Plate-forme	Cette police signale une plate-forme spécifique et indique que le texte qui suit s'applique uniquement à cette plate-forme.
clic droit	(Macintosh) Appuyez <Command>-cliquez pour effectuer la même action qu'un clic droit.

Concepts LabVIEW

Cette partie décrit les concepts de programmation pour la construction d'applications dans LabVIEW. Les chapitres de cette section vous présentent l'environnement de programmation de LabVIEW et vous aident à concevoir votre application.

La partie I, *Concepts LabVIEW*, comprend les chapitres suivants :

- Le chapitre 1, *Introduction à LabVIEW*, décrit LabVIEW, la documentation, fournie et complète, ainsi que les outils qui vous aident à concevoir et à construire les VIs.
- Le chapitre 2, *Introduction aux instruments virtuels*, décrit les composants des instruments virtuels, appelés VIs.
- Le chapitre 3, *Environnement LabVIEW*, décrit les palettes, les outils et les menus de LabVIEW que vous utilisez pour construire les faces-avant et les diagrammes des VIs. Ce chapitre explique aussi comment personnaliser les palettes de LabVIEW et définir plusieurs options d'environnement de travail.
- Le chapitre 4, *Construction de la face-avant*, explique comment construire la face-avant d'un VI.
- Le chapitre 5, *Construction du diagramme*, explique comment construire le diagramme d'un VI.
- Le chapitre 6, *Exécution et mise au point des VIs*, explique comment configurer l'exécution d'un VI et identifier les problèmes d'organisation du diagramme ou de transfert des données dans le diagramme.
- Le chapitre 7, *Création de VIs et de sous-VIs*, explique comment créer vos propres VIs et sous-VIs, répartir les VIs et construire des applications ou des bibliothèques partagées autonomes.

Introduction à LabVIEW

LabVIEW est un langage de programmation graphique qui utilise des icônes et non des lignes de texte pour créer des applications. À l'inverse des langages de programmation textuels dans lesquels des instructions déterminent l'exécution du programme, LabVIEW utilise une programmation par flux de données, ce flux déterminant l'exécution.

Sous LabVIEW, vous construisez une interface utilisateur en utilisant un ensemble d'outils et d'objets. L'interface utilisateur d'un VI est appelée la face-avant. Ensuite, vous créez le code en utilisant des représentations graphiques de fonctions pour contrôler les objets de la face-avant. Le diagramme contient ce code. D'une certaine manière, le diagramme ressemble à un organigramme.

Vous pouvez acheter plusieurs toolsets optionnels pour développer des applications spécialisées. Tous les toolsets s'intègrent parfaitement dans LabVIEW. Reportez-vous au site de National Instruments à l'adresse ni.com si vous souhaitez plus d'informations sur ces toolsets.

Ressources de documentation LabVIEW

LabVIEW inclut une vaste documentation pour les utilisateurs LabVIEW novices ou confirmés. Les manuels LabVIEW et les notes d'application sont également disponibles au format PDF. Vous devez également avoir installé Adobe Acrobat Reader, version 4.0 ou ultérieure, pour lire les fichiers PDF. Reportez-vous au site Web d'Adobe Systems Incorporated à l'adresse www.adobe.com pour télécharger Acrobat Reader. Reportez-vous à la bibliothèque des manuels des produits National Instruments à l'adresse ni.com pour obtenir des ressources de documentation mises à jour.

- **Bibliothèque LabVIEW** : utilisez cette bibliothèque au format PDF pour rechercher des versions PDF de tous les manuels et notes d'application LabVIEW. Accédez à la *Bibliothèque LabVIEW* en sélectionnant **Aide»Rechercher dans les manuels PDF...**
- **Initiation à LabVIEW** : utilisez ce manuel pour vous familiariser avec l'environnement de programmation graphique LabVIEW et avec les fonctionnalités LabVIEW de base dont vous avez besoin pour

construire des applications de contrôle d'acquisition de données et de contrôle d'instrument.

- **Tutorial LabVIEW** : utilisez ce tutorial pour apprendre les concepts de base LabVIEW. Le tutorial vous guide à travers plusieurs activités qui vous aident à vous familiariser avec la programmation graphique. Accédez au *Tutorial LabVIEW* en sélectionnant **Aide»Aide LabVIEW...** ou en cliquant sur le bouton **Tutorial LabVIEW** dans la boîte de dialogue **LabVIEW**.
- **Carte de référence rapide de LabVIEW** : utilisez cette carte pour commencer rapidement à utiliser LabVIEW. La carte décrit les palettes LabVIEW et les techniques générales d'édition, de câblage et de mise au point.
- **Manuel de l'utilisateur LabVIEW** : utilisez ce manuel pour vous familiariser avec les concepts de programmation, les techniques, les fonctionnalités, les VIs et fonctions dont vous avez besoin pour créer des applications de test et de mesure, d'acquisition de données, de contrôle d'instrument, de fichier journal, d'analyse de mesure et de génération de rapports LabVIEW.
- **Aide LabVIEW** : utilisez ce fichier d'aide comme référence pour obtenir des informations sur les palettes, menus, outils, VIs et fonctions de LabVIEW. L'*Aide LabVIEW* inclut également des instructions pas à pas pour utiliser les fonctionnalités LabVIEW. Accédez à l'*Aide LabVIEW* en sélectionnant **Aide»Aide LabVIEW...**

L'*Aide LabVIEW* inclut des liens vers les ressources suivantes :

- *Tutorial LabVIEW*
- *Bibliothèque LabVIEW*, qui inclut des versions PDF des manuels et des notes d'application LabVIEW.
- Ressources de support technique sur le site Web de National Instruments, telles que la page Developer Zone, la base de connaissances (voir KnowledgeBase) et la bibliothèque des manuels sur les produits (voir Product Manuals Library)



Remarque (Macintosh et UNIX) National Instruments vous recommande d'utiliser Netscape 6.0 ou version ultérieure pour visualiser l'*Aide LabVIEW*.

- **LabVIEW Measurements Manual** : utilisez ce manuel pour en savoir plus sur la construction des applications d'acquisition de données et de contrôle d'instrument dans LabVIEW. Si vous êtes un utilisateur novice de LabVIEW, lisez le document *Initiation à LabVIEW* et le *Manuel de l'utilisateur LabVIEW* avant de lire ce manuel.

- **LabVIEW Development Guidelines** : utilisez ce manuel pour apprendre à construire des VIs faciles à comprendre, à utiliser et à modifier. Ce manuel décrit les techniques de traçage, de conception et de documentation.



Remarque Le manuel *LabVIEW Development Guidelines* est disponible uniquement dans le Système de développement professionnel LabVIEW. La version PDF est disponible dans toutes les versions de LabVIEW.

- **Initiation aux VIs d'analyse point par point** : utilisez ce guide pour vous apprendre à utiliser les VIs d'analyse point par point de LabVIEW.
- **Using External Code in LabVIEW** : utilisez ce manuel pour apprendre à utiliser le Code Interface Node (CIN) et les sous-programmes externes pour importer un code écrit dans des langages de programmation textuels. Ce manuel inclut des informations sur les sous-programmes, les bibliothèques de fonctions, les routines de manipulation de la mémoire et des fichiers et les routines de diagnostic, qui sont externes et partagés. Ce manuel comprend également des informations sur l'appel des DLL.



Remarque Le manuel *Using External Code in LabVIEW* est disponible uniquement au format PDF.

- **Notes d'application LabVIEW** : utilisez les notes d'application LabVIEW pour vous familiariser avec les concepts et applications avancés ou spécialisés de LabVIEW. Reportez-vous à la page NI Developer Zone à l'adresse ni.com pour prendre connaissance des notes d'application les plus récentes.
- **LabVIEW VXI VI Reference Manual** : utilisez ce manuel pour vous familiariser avec les VIs VXI pour LabVIEW. Ce manuel accompagne le *NI-VXI Programmer Reference Manual* qui accompagne le matériel VXI.



Remarque Le *LabVIEW VXI VI Reference Manual* est disponible uniquement au format PDF.

VIs d'exemple et outils LabVIEW

Utilisez les VIs d'exemple et les outils LabVIEW pour vous aider à créer et à construire des VIs.

VIs d'exemple LabVIEW

LabVIEW inclut des centaines de VIs d'exemple que vous pouvez utiliser et incorporer dans vos propres VIs. Vous pouvez modifier un exemple pour qu'il corresponde à votre application ou vous pouvez copier et coller des éléments à partir de plusieurs exemples à l'intérieur de votre propre VI. Parcourez ou recherchez les VIs d'exemple en sélectionnant **Aide» Recherche d'exemples**. Reportez-à la page NI Developer Zone à l'adresse ni.com pour prendre connaissance de VIs d'exemple supplémentaires.

Outils LabVIEW

LabVIEW comprend de nombreux outils qui vous aident à configurer rapidement vos instruments de mesure, y compris les suivants. Vous pouvez accéder à ces outils à partir du menu **Outils**.

- **(Windows)** Measurement & Automation Explorer vous aide à configurer vos logiciels et matériel National Instruments.
- **(Macintosh)** L'utilitaire de configuration NI-DAQ vous aide à configurer le matériel DAQ de National Instruments.
- **(Macintosh)** L'Assistant Entrées/Sorties DAQ vous aide à définir le type de périphérique connecté aux voies du matériel DAQ. Après avoir défini une voie, l'Assistant Entrées/Sorties DAQ stocke les paramètres en mémoire.
- **(Windows et Macintosh)** L'afficheur de voies virtuelles répertorie les voies DAQ configurées.
- **(Windows et Macintosh)** L'Assistant Solutions DAQ vous aide à trouver des solutions pour des applications DAQ courantes. Vous pouvez choisir parmi des VIs d'exemple ou créer des VIs personnalisés.

Introduction aux instruments virtuels

Les programmes LabVIEW sont appelés instruments virtuels ou VIs, leur aspect et leur fonctionnement imitant des instruments réels, tels que des oscilloscopes et des multimètres. Chaque VI utilise des fonctions qui manipulent les entrées à partir de l'interface utilisateur ou d'autres sources et affiche ou déplace ces informations vers d'autres fichiers ou ordinateurs.

Un VI contient les trois composantes suivantes :

- **Face-avant** : sert d'interface utilisateur.
- **Diagramme** : contient le code source graphique qui définit les fonctionnalités du VI.
- **Icône et cadre connecteur** : identifie le VI pour que vous puissiez l'utiliser dans un autre VI. Un VI à l'intérieur d'un autre VI est appelé sous-VI. Un sous-VI correspond à un sous-programme dans des langages de programmation textuels.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour de plus amples informations sur la création de VIs et de sous-VIs.

Face-avant

La face-avant est l'interface utilisateur du VI. La figure 2-1 montre un exemple de face-avant.

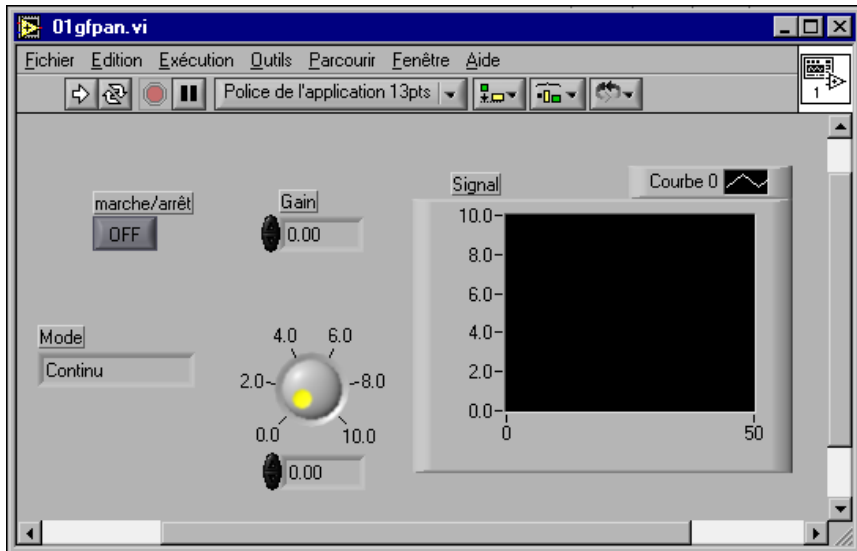


Figure 2-1. Exemple de face-avant

Vous construisez la face-avant à l'aide de commandes et d'indicateurs, qui sont respectivement les terminaux d'entrées et de sorties interactifs du VI. Les commandes sont des boutons rotatifs, des boutons-poussoirs, des cadrans et autres dispositifs d'entrée. Les indicateurs sont des graphes, des LED et autres afficheurs. Les commandes simulent les périphériques d'entrée d'instruments et fournissent les données au diagramme du VI. Les indicateurs simulent les dispositifs de sortie d'instruments et affichent les données que le diagramme acquiert ou génère. Reportez-vous au chapitre 4, *Construction de la face-avant*, pour de plus amples informations sur la face-avant.

Diagramme

Après avoir construit la face-avant, vous devez ajouter le code en utilisant les représentations graphiques des fonctions pour contrôler les objets de la face-avant. Le diagramme contient ce code source graphique. Les objets de la face-avant apparaissent comme des terminaux sur le diagramme.

Reportez-vous au chapitre 5, *Construction du diagramme*, pour de plus amples informations sur le diagramme.

Le VI de la figure 2-2 montre plusieurs objets élémentaires du diagramme, tels que des terminaux, fonctions et fils de liaison.

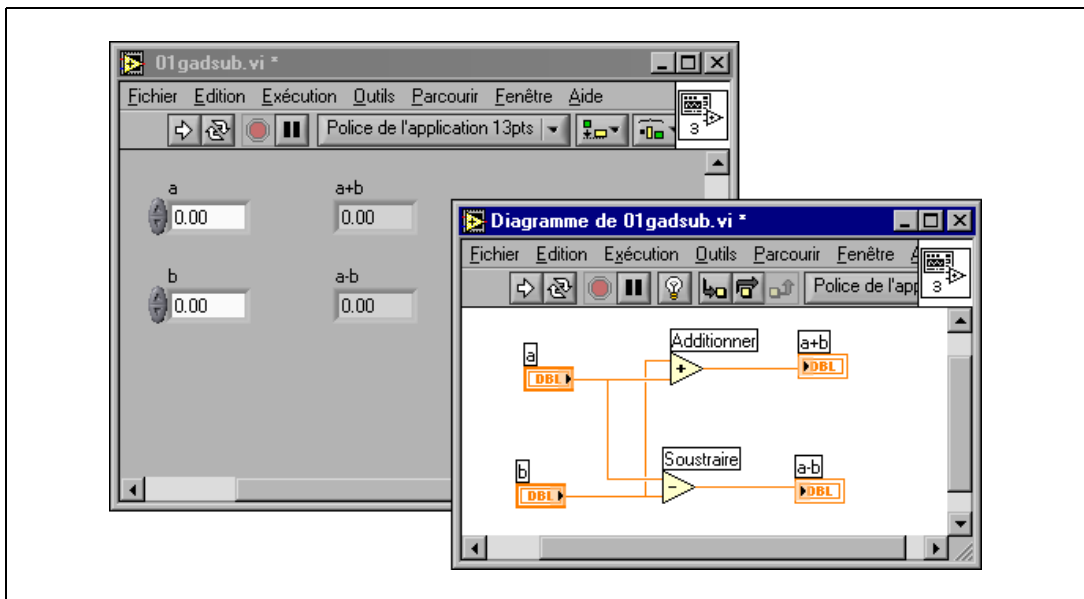


Figure 2-2. Exemple de diagramme et face-avant correspondante

Terminaux



Les terminaux représentent le type de données de la commande ou de l'indicateur. Par exemple, un terminal DBL, tel que celui affiché à gauche, représente une commande ou un indicateur numérique à virgule flottante double précision. Reportez-vous à la section *Type de données des indicateurs et des commandes* du chapitre 5, *Construction du diagramme*, pour de plus amples informations sur les types de données LabVIEW et leurs représentations graphiques.

Les terminaux correspondent à des ports d'entrée et de sortie qui échangent des informations entre la face-avant et le diagramme. Les données que vous entrez dans les commandes de la face-avant (**a** et **b** sur la figure 2-2) sont transmises au diagramme par le biais des terminaux de commande. Les données rentrent ensuite dans les fonctions Ajouter et Soustraire. Lorsque les fonctions Ajouter et Soustraire terminent leurs calculs internes, elles génèrent de nouvelles valeurs de données. Les données s'acheminent vers les terminaux d'indicateurs, où elles quittent le diagramme, puis réintègrent la face-avant en apparaissant dans les indicateurs de celle-ci.

Nœuds

Les nœuds sont des objets du diagramme qui possèdent des entrées et/ou des sorties et qui effectuent des opérations lorsqu'un VI s'exécute. Ils sont analogues aux déclarations, opérateurs, fonctions et sous-programmes écrits dans des langages de programmation textuels. Les fonctions Ajouter et Soustraire de la figure 2-2 correspondent à des nœuds. Reportez-vous à la section *Nœuds du diagramme* du chapitre 5, *Construction du diagramme*, pour de plus amples informations à propos des nœuds.

Fils de liaison

Vous transférez les données entre les objets du diagramme à l'aide des fils de liaison. Dans la figure 2-2, les fils de liaison connectent les terminaux DBL des commandes et indicateurs aux fonctions Ajouter et Soustraire. Chaque fil de liaison a une source de données unique, mais vous pouvez le câbler à de nombreux VIs et fonctions qui lisent les données. Les fils de liaison sont de couleur, style et épaisseur différents selon leur type de données. Un fil de liaison brisé apparaît comme une ligne en pointillé. Reportez-vous à la section *Utilisation des fils de liaison pour relier les objets du diagramme* du chapitre 5, *Construction du diagramme*, pour de plus amples informations à propos des fils de liaison.

Structures

Les structures sont les représentations graphiques des boucles et des déclarations de cas des langages de programmation textuels. Utilisez les structures du diagramme pour répéter des blocs de code et exécuter un code en fonction d'une certaine condition ou dans un ordre spécifique. Reportez-vous au chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour obtenir des exemples et de plus amples informations sur les structures.

Icône et connecteur



Après avoir construit la face-avant et le diagramme d'un VI, vous devez construire l'icône et le connecteur pour pouvoir utiliser le VI comme un sous-VI. Chaque VI affiche une icône, telle que celle représentée à gauche, dans le coin supérieur droit des fenêtres de la face-avant et du diagramme. Une icône est la représentation graphique d'un VI. Elle peut contenir du texte, des images ou une combinaison des deux. Si vous utilisez un VI comme sous-VI, l'icône identifie le sous-VI sur le diagramme du VI. Vous pouvez double-cliquer sur l'icône pour la personnaliser ou la modifier. Reportez-vous à la section [Création d'une icône](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour de plus amples informations sur les icônes.



Vous devez également construire un connecteur, représenté à gauche, pour utiliser le VI comme sous-VI. Le connecteur est un groupe de terminaux correspondant aux commandes et aux indicateurs de ce VI ; ce groupe est semblable à la liste de paramètres d'un appel de fonction dans les langages de programmation textuels. Le connecteur définit les entrées et sorties que vous pouvez connecter au VI que vous voulez utiliser comme sous-VI. Un connecteur reçoit des données sur ses terminaux d'entrée et transmet les données au diagramme par les commandes de sa face-avant. Il reçoit les résultats sur ses terminaux de sortie par les indicateurs de sa face-avant.

Lorsque vous affichez le connecteur pour la première fois, vous voyez un modèle de connecteur. Vous pouvez sélectionner un modèle différent si vous le souhaitez. En général, le connecteur possède un terminal pour chaque commande ou indicateur sur la face-avant. Vous pouvez définir jusqu'à 28 terminaux sur un connecteur. Si vous prévoyez d'apporter des modifications au VI nécessitant une nouvelle entrée ou sortie, laissez des terminaux supplémentaires non définis. Reportez-vous à la section [Configuration du connecteur](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour de plus amples informations sur la configuration des connecteurs.



Remarque Évitez d'attribuer plus de 16 terminaux à un VI. Trop de terminaux risquent de réduire la lisibilité et la facilité d'emploi du VI.

Utilisation et personnalisation des VIs et sous-VIs

Après avoir construit un VI et créé ses icônes et connecteur, vous pouvez l'utiliser comme un sous-VI. Reportez-vous à la section [Sous-VIs](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour de plus amples informations sur les sous-VIs.

Vous pouvez enregistrer les VIs comme fichiers individuels ou regrouper plusieurs VIs ensemble pour ensuite les enregistrer dans une bibliothèque de VIs. Reportez-vous à la section [Enregistrement des VIs](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour de plus amples informations sur l'enregistrement des VIs dans des bibliothèques.

Vous pouvez personnaliser l'aspect et le comportement d'un VI. Vous pouvez également créer des menus personnalisés pour chaque VI que vous construisez et les configurer pour afficher ou masquer les barres de menus. Reportez-vous au chapitre 15, [Personnalisation des VIs](#), pour de plus amples informations sur la manière de personnaliser un VI.

Environnement LabVIEW

Utilisez les palettes, les outils et les menus LabVIEW pour construire les faces-avant et les diagrammes des VIs. Vous pouvez personnaliser les palettes de **Commandes** et de **Fonctions** et définir plusieurs options d'environnement de travail.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur l'utilisation des palettes, des menus et des barres d'outils et sur la manière de personnaliser votre environnement de travail.

Palette de commandes

La palette de **Commandes** est disponible uniquement sur la face-avant. Cette palette contient les commandes et les indicateurs que vous utilisez pour construire la face-avant. Les commandes et les indicateurs sont situés dans des sous-palettes suivant leur type. Reportez-vous à la section *Commandes et indicateurs de la face-avant* du chapitre 4, *Construction de la face-avant* pour de plus amples informations sur les types de commande et d'indicateur.

Sélectionnez **Fenêtre»Palette de commandes** ou faites un clic droit sur l'espace de travail de la face-avant pour afficher la palette de **Commandes**. Vous pouvez placer la palette de **Commandes** n'importe où sur l'écran.

Vous pouvez modifier l'apparence de la palette de **Commandes**. Reportez-vous à la section *Personnalisation des palettes de commandes et de fonctions* de ce chapitre pour de plus amples informations sur la personnalisation de la palette de **Commandes**.

Palette de fonctions

La palette de **Fonctions** est disponible uniquement sur le diagramme. Cette palette contient les VIs et les fonctions que vous utilisez pour construire le diagramme. Les VIs et les fonctions sont situés dans des sous-palettes

suivant leur type. Reportez-vous à la section [Sommaire des fonctions](#) du chapitre 5, [Construction du diagramme](#) pour de plus amples informations sur les types de VIs et de fonctions.

Sélectionnez **Fenêtre»Palette de fonctions** ou faites un clic droit sur l'espace de travail du diagramme pour afficher la palette de **Fonctions**. Vous pouvez placer la palette de **Fonctions** n'importe où sur l'écran.

Vous pouvez modifier l'apparence de la palette de **Fonctions**.

Reportez-vous à la section [Personnalisation des palettes de commandes et de fonctions](#) de ce chapitre pour obtenir de plus amples informations sur la personnalisation de la palette de **Fonctions**.

Parcourir les palettes de Commandes et de Fonctions

Utilisez les boutons de navigation situés sur les palettes de **Commandes** et de **Fonctions** pour parcourir et rechercher des commandes, des VIs et des fonctions. Lorsque vous cliquez sur l'icône d'une sous-palette, la palette entière devient la sous-palette que vous avez sélectionnée. Vous pouvez aussi cliquer avec le bouton droit sur l'icône d'un VI dans la palette et sélectionner **Ouvrir un VI** à partir du menu local pour y parvenir.

Les palettes de **Commandes** et de **Fonctions** contiennent les boutons de navigation suivants :



- **Remonter l'arborescence** : vous fait passer à un niveau supérieur dans la hiérarchie de la palette.



- **Rechercher** : met la palette en mode de recherche. En mode de recherche, vous pouvez effectuer des recherches textuelles pour trouver des commandes, des VIs ou des fonctions dans les palettes.



- **Options** : ouvre la boîte de dialogue des **Options du navigateur de fonctions** à partir de laquelle vous pouvez configurer l'apparence des palettes.

Palette d'outils

La palette d'**Outils** est disponible sur la face-avant et le diagramme. Un outil est un mode de fonctionnement spécial du curseur de la souris. Le curseur correspond à l'icône de l'outil sélectionné dans la palette. Utilisez les outils pour faire fonctionner et modifier les objets de la face-avant et du diagramme.

Sélectionnez **Fenêtre»Palette d'outils** pour afficher la palette d'**Outils**. Vous pouvez placer cette palette n'importe où sur l'écran.



Conseil Appuyez sur la touche <Maj> et faites un clic droit pour afficher une version provisoire de la palette d'**Outils** à l'emplacement du curseur.

Si la sélection automatique de l'outil est activée et que vous déplacez le curseur sur les objets de la face-avant ou du diagramme, LabVIEW sélectionne automatiquement l'outil correspondant dans la palette d'**Outils**. Activez la sélection automatique de l'outil en cliquant sur le bouton **Sélection automatique de l'outil** dans la palette d'**Outils** ou en appuyant sur les touches <Maj-Tab>.

Les menus et la barre d'outils

Utilisez les éléments des menus et de la barre d'outils pour faire fonctionner et modifier les objets de la face-avant et du diagramme. Utilisez les boutons de la barre d'outils pour exécuter les VIs.

Menus

Les menus situés en haut de la fenêtre d'un VI contiennent les éléments communs à d'autres applications, par exemple **Ouvrir**, **Enregistrer**, **Copier** et **Coller**, ainsi que d'autres éléments propres à LabVIEW. Certains éléments de menus donnent également des raccourcis clavier.

(Macintosh) Les menus apparaissent en haut de l'écran.



Remarque Certains éléments de menus ne sont pas disponibles lorsqu'un VI est en mode Exécution.

Menus locaux

Le menu le plus souvent utilisé est le menu local de l'objet. Dans LabVIEW, tous les objets ainsi que l'espace vide sur la face-avant et le diagramme possèdent des menus locaux associés. Utilisez les éléments des menus locaux pour modifier l'apparence ou le comportement des objets de la face-avant et du diagramme. Pour accéder au menu local, faites un clic droit sur l'objet, la face-avant ou le diagramme.

(Macintosh) Appuyez sur la touche <Command> et faites un clic sur l'objet, la face-avant ou le diagramme.

Menus locaux en mode Exécution

Lorsqu'un VI est en cours d'exécution ou en mode Exécution, tous les objets de la face-avant disposent d'un menu local comprenant un ensemble réduit d'éléments. Utilisez les éléments du menu local réduit pour couper, copier ou coller le contenu de l'objet, pour définir la valeur par défaut de l'objet ou pour en lire sa description.

Certaines commandes complexes ont des options supplémentaires. Par exemple, le menu local tableau comprend des éléments servant à copier une gamme de valeurs ou atteindre le dernier élément du tableau.

Barre d'outils

Utilisez les boutons de la barre d'outils pour exécuter et modifier un VI. Lorsque vous exécutez un VI, des boutons apparaissent sur la barre d'outils que vous pouvez utiliser pour mettre le VI au point.

Personnalisation de votre environnement de travail

Vous pouvez modifier la manière dont les palettes de **Commandes** et de **Fonctions** apparaissent à l'écran et utiliser la boîte de dialogue **Options** pour paramétrer les autres options de votre environnement de travail.

Personnalisation des palettes de commandes et de fonctions

Vous pouvez personnaliser les palettes de **Commandes** et de **Fonctions** de différentes façons :

- Ajouter des VIs et des commandes aux palettes.
- Paramétrer des visualisations différentes en fonction des utilisateurs, cacher certains VIs et fonctions pour rendre LabVIEW plus convivial pour quelques utilisateurs tout en fournissant l'intégralité des palettes à d'autres utilisateurs.
- Disposer les palettes intégrées de façon à faciliter l'accès aux VIs et fonctions que vous utilisez le plus fréquemment.
- Convertir un ensemble de commandes ActiveX en commandes personnalisées, puis les ajouter aux palettes.
- Ajouter des toolsets aux palettes.

Ajouter des VIs et des commandes à la bibliothèque utilisateur et à la bibliothèque d'instruments

Pour ajouter des VIs et des commandes aux palettes de **Commandes** et de **Fonctions**, la méthode la plus simple consiste à les enregistrer dans le répertoire `user.lib`. Lorsque vous redémarrez LabVIEW, les palettes **Fonctions»Bibliothèques utilisateur** et **Commandes»Commandes utilisateur** contiennent les sous-palettes de chaque répertoire, un fichier bibliothèque de VIs (`.lib`) ou menu (`.mnu`) dans `user.lib` et les icônes de chaque fichier dans `user.lib`. LabVIEW met automatiquement à jour les palettes lorsque vous redémarrez LabVIEW après avoir ajouté ou retiré des fichiers provenant de répertoires spécifiques.

La palette **Fonctions»E/S d'instruments»Drivers d'instruments** correspond au répertoire `instr.lib`. Enregistrez les drivers d'instruments dans ce répertoire pour pouvoir y accéder facilement à partir de la palette de **Fonctions**.

Lorsque vous ajoutez des VIs ou des commandes aux palettes de **Commandes** et de **Fonctions** en suivant cette méthode, vous ne pouvez pas déterminer l'emplacement exact des VIs ou des commandes dans les palettes.

Création et modification de l'apparence d'une palette

Pour contrôler l'emplacement exact des VIs et des commandes que vous ajoutez dans les palettes de **Commandes** et de **Fonctions**, vous devez créer une vue de la palette. LabVIEW intègre quatre types de vues : standard, élémentaire, acquisition de données et test & mesure.

LabVIEW stocke les informations concernant les palettes de **Commandes** et de **Fonctions** dans le répertoire `labview\menus`. Le répertoire `menus` contient les répertoires correspondant à chaque vue que vous avez créée ou installée. Si vous utilisez LabVIEW en réseau, vous pouvez définir des répertoires `menus` individuels pour chaque utilisateur, ce qui facilite le transfert des vues à d'autres personnes.

Lorsque vous créez une nouvelle vue d'une palette, LabVIEW utilise une copie de la vue par défaut initialement présente, sur laquelle vous pouvez alors introduire n'importe quel changement. LabVIEW copie la palette intégrée d'origine située dans le répertoire `labview\menus` avant l'introduction de toute modification. Cette protection des palettes intégrées permet de faire des essais avec les palettes sans altérer la vue initiale.

Comment LabVIEW enregistre les vues

Les fichiers `.mnu` et `.lib` contiennent chacun une palette de **Commandes** et une palette de **Fonctions**. De plus, chaque fichier contient une icône pour les palettes de **Commandes** et de **Fonctions**. Vous devez enregistrer chaque sous-palette que vous créez dans un fichier `.mnu` distinct.

Lorsque vous sélectionnez une vue, LabVIEW recherche le sous-répertoire correspondant à cette vue dans le répertoire `menus`. Il construit les palettes de **Commandes** et de **Fonctions** de niveau principal, ainsi que les sous-palettes, à partir du fichier `root.mnu` situé dans ce sous-répertoire que LabVIEW crée automatiquement chaque fois que vous créez une vue.

LabVIEW crée une icône sur la palette pour chaque VI ou commande. Pour chaque sous-répertoire, fichier `.mnu` ou fichier `.lib`, LabVIEW crée une sous-palette dans la palette.

Construction des sous-palettes ActiveX

Si vous utilisez des contrôles ActiveX sur la face-avant, sélectionnez **Outils»Avancé»Importer des contrôles ActiveX...** pour convertir un ensemble de contrôles ActiveX en commandes personnalisées et les ajouter à la palette de **Commandes**. Par défaut, LabVIEW enregistre les commandes dans le répertoire `user.lib` car tous les fichiers et répertoires de `user.lib` apparaissent automatiquement dans les palettes.

Représentation des toolsets dans les palettes

Les toolsets que vous installez dans `vi.lib\addons` apparaissent automatiquement au niveau principal des palettes de **Commandes** et de **Fonctions** après le redémarrage de LabVIEW. Si des toolsets ont été installés ailleurs, vous pouvez les déplacer jusqu'au répertoire `addons` pour pouvoir y accéder plus facilement.



Mise en garde N'enregistrez *pas* vos propres VIs et commandes dans le répertoire `vi.lib` car LabVIEW écrase ces fichiers à chaque mise à jour. Enregistrez vos VIs et vos commandes dans le répertoire `user.lib` pour les ajouter aux palettes de **Commandes** et de **Fonctions**.

Paramétrage des options de l'environnement de travail

Sélectionnez **Outils»Options...** pour personnaliser LabVIEW. Utilisez la boîte de dialogue **Options** afin de définir les options pour les chemins, les performances et problèmes de disque, les faces-avant, les diagrammes, l'opération d'annulation, les outils de mise au point, les couleurs, les

polices, l'impression, la fenêtre **Historique**, les formats de l'heure et de la date et les autres fonctionnalités de LabVIEW.

Utilisez le menu déroulant du haut de la boîte de dialogue **Options** pour effectuer votre sélection parmi les diverses catégories.

Comment LabVIEW enregistre les options

Vous n'avez pas à modifier manuellement les options, ni à connaître leur format exact car la boîte de dialogue **Options** le fait à votre place. LabVIEW enregistre les options de manière différente pour chaque plate-forme.

Windows

LabVIEW enregistre les options dans le fichier `labview.ini` du répertoire LabVIEW. Le format de ce fichier est similaire à celui des autres fichiers `.ini`. Il commence par un repère de section LabVIEW suivi par le nom de l'option et par ses valeurs, telles que `offscreenUpdates=True`.

Si vous désirez utiliser un autre fichier d'options, spécifiez-le dans le raccourci que vous utilisez pour démarrer LabVIEW. Par exemple, pour utiliser un fichier d'options enregistré dans votre ordinateur sous le nom de `lvrc` au lieu de `labview.ini`, cliquez avec le bouton droit de la souris sur l'icône LabVIEW du bureau puis sélectionnez **Propriétés**. Cliquez sur l'onglet **Raccourci** et tapez `labview -pref lvrc` dans la boîte de texte **Cible**.

Macintosh

LabVIEW stocke les options dans le fichier texte `LabVIEW Preferences` du répertoire **System»Preferences**.

Si vous désirez utiliser un autre fichier d'options, copiez le fichier `LabVIEW Preferences` dans le dossier **LabVIEW**, puis modifiez les options dans la boîte de dialogue **Options**. Lorsque vous redémarrez LabVIEW, il commence par rechercher le fichier d'options dans le dossier **LabVIEW**. S'il n'y trouve pas le fichier, il le recherche dans le dossier **System**. S'il n'y trouve pas le fichier, il en crée un nouveau dans le dossier **System**. LabVIEW écrit toutes les modifications que vous avez apportées au niveau de la boîte de dialogue **Options** dans le premier fichier `LabVIEW Preferences` qu'il trouve.

UNIX

LabVIEW enregistre les options dans le fichier `.labviewrc` de votre répertoire racine. Si vous modifiez une option dans la boîte de dialogue **Options**, LabVIEW écrit ces modifications dans le fichier `.labviewrc`. Vous pouvez créer un fichier `labviewrc` dans le répertoire du programme afin d'y stocker les options qui sont les mêmes pour tous les utilisateurs, telles que le chemin de recherche de VI. Utilisez le fichier `.labviewrc` pour enregistrer les options qui diffèrent en fonction de l'utilisateur, telles que les paramètres de police ou de couleurs, car les entrées du fichier `.labviewrc` de votre répertoire racine écrasent les entrées du répertoire du programme avec lesquelles elles sont en conflit.

Par exemple, si vous avez installé les fichiers LabVIEW dans `/opt/labview`, LabVIEW commence par lire les options à partir de `/opt/labview/labviewrc`. Si vous modifiez une option dans la boîte de dialogue **Options**, telle que la police de l'application, LabVIEW écrit cette modification dans le fichier `.labviewrc`. Lorsque LabVIEW redémarre, il utilise l'option de la police de l'application du fichier `.labviewrc` et non la police de l'application établie par défaut dans `/opt/labview/labviewrc`.

Les entrées des options sont constituées du nom de l'option suivi par une virgule et une valeur. Le nom de l'option est l'exécutable suivi par un point (.) et une option. Lorsque LabVIEW recherche des noms d'option, il différencie les majuscules des minuscules. Vous pouvez entourer la valeur d'une option par des guillemets simples ou doubles. Par exemple, pour utiliser une précision double par défaut, ajoutez l'entrée suivante au fichier `.labviewrc` de votre répertoire racine.

```
labview.defPrecision : double
```

Si vous désirez utiliser un autre fichier d'options, spécifiez-le dans la ligne de commande utilisée pour démarrer LabVIEW. Par exemple, pour utiliser un fichier nommé `lvrc` du répertoire `test` au lieu du fichier `.labviewrc`, tapez `labview -pref /test/lvrc`. LabVIEW écrit toutes les modifications que vous avez effectuées au niveau de la boîte de dialogue **Options** dans le fichier d'options `lvrc`. Lorsque vous spécifiez un fichier d'options dans la ligne de commande, LabVIEW lit tout de même le fichier `labviewrc` du répertoire du programme, mais les entrées du fichier d'options que vous avez spécifiées dans la ligne de commande écrasent les entrées du répertoire du programme avec lesquelles elles sont en conflit.

Construction de la face-avant

La face-avant est l'interface utilisateur d'un VI. En général, vous devez commencer par concevoir la face-avant avant de concevoir le diagramme nécessaire pour effectuer les tâches sur les entrées et les sorties créées avec la face-avant. Reportez-vous au chapitre 5, *Construction du diagramme*, pour de plus amples informations sur le diagramme.

Vous construisez la face-avant à l'aide de commandes et d'indicateurs, qui sont respectivement les terminaux d'entrées et de sorties interactifs du VI. Les commandes sont des boutons rotatifs, des boutons-poussoirs, des cadrans et autres dispositifs d'entrée. Les indicateurs sont des graphes, des LED et autres afficheurs. Les commandes simulent les dispositifs d'entrée d'instruments et fournissent les données au diagramme du VI. Les indicateurs simulent les dispositifs de sortie d'instruments et affichent les données que le diagramme acquiert ou génère.

Sélectionnez **Fenêtre»Palette de commandes** pour afficher la palette de **Commandes**, puis sélectionnez les commandes et indicateurs à partir de celle-ci et placez-les sur la face-avant.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour de plus amples informations sur la conception et la configuration de la face-avant.

Configuration des objets de la face-avant

Vous pouvez personnaliser la face-avant en utilisant les menus locaux pour les commandes et les indicateurs, en établissant l'ordre des objets de la face-avant et en utilisant des graphiques importés. Vous pouvez également redimensionner manuellement les objets de la face-avant et les paramétrer pour qu'ils se redimensionnent automatiquement lorsque la taille de la fenêtre change. Reportez-vous à la note d'application *LabVIEW Custom Controls, Indicators, and Type Definitions* pour de plus amples informations sur la création et l'utilisation de commandes, d'indicateurs et de définitions de type personnalisés.

Afficher et masquer des éléments optionnels

Les commandes et indicateurs de la face-avant possèdent des éléments optionnels que vous pouvez afficher ou masquer. Affichez la liste des éléments disponibles en cliquant avec le bouton droit de la souris sur un objet et en sélectionnant **Éléments visibles** dans le menu local. La plupart des objets ont une étiquette et un sous-titre. Reportez-vous à la section [Étiquetage](#) de ce chapitre pour obtenir de plus amples informations sur les étiquettes et les sous-titres.

Changement des commandes en indicateurs et vice versa

LabVIEW configure initialement les objets de la palette des **Commandes** en tant que commandes ou indicateurs en fonction de leur usage principal. Par exemple, si vous sélectionnez un commutateur sur la palette **Commandes»Booléen**, il apparaît sur la face-avant en tant que commande car les commutateurs sont utilisés le plus souvent comme périphériques d'entrée. Si vous sélectionnez une LED, elle apparaît sur la face-avant comme un indicateur car les LED sont utilisées le plus souvent comme dispositifs de sortie.

Certaines palettes contiennent une commande et un indicateur pour le même type ou la même classe d'objet. Par exemple, la palette **Commandes»Numérique** contient une commande numérique et un indicateur numérique.

Vous pouvez changer les commandes en indicateurs et vice versa en effectuant un clic droit sur l'objet puis en sélectionnant **Changer en commande** ou **Changer en indicateur** dans le menu local.

Remplacement des objets de la face-avant

Vous pouvez remplacer un objet de la face-avant par une commande ou un indicateur différent. Lorsque vous cliquez sur un objet avec le bouton droit et sélectionnez **Remplacer** dans le menu local, une palette de **Commandes** temporaire apparaît et ce, même si la palette de **Commandes** est déjà ouverte. Sélectionnez une commande ou un indicateur dans la palette de **Commandes** temporaire pour remplacer l'objet actuel de la face-avant.

La sélection de **Remplacer** dans le menu local permet de conserver autant d'informations que possible sur l'objet d'origine, telles que son nom, sa description, ses données par défaut, le sens de flux des données (commande ou indicateur), sa couleur, sa taille, etc. Cependant, le nouvel objet conserve son propre type de données. Les fils de liaison du terminal de l'objet ou des variables locales demeurent dans le diagramme, mais ils sont parfois brisés.

Par exemple, si vous remplacez un terminal numérique par un terminal de chaîne de caractères, le fil de liaison d'origine demeure dans le diagramme, mais il est brisé.

Plus le nouvel objet ressemble à l'objet que vous venez de remplacer et plus vous pouvez conserver ses caractéristiques d'origine. Par exemple, si vous remplacez une glissière par une glissière d'un style différent, la nouvelle glissière conserve les mêmes hauteur, échelle, valeur, nom, description, etc. Si vous remplacez en revanche cette glissière par une commande de chaîne de caractères, LabVIEW ne conserve que le nom, la description et le sens du flux des données car une glissière et une commande de chaîne de caractères n'ont pas grand-chose en commun.

Vous pouvez aussi coller des objets provenant du presse-papiers pour remplacer des commandes ou des indicateurs existants de la face-avant. Cette méthode ne permet de conserver aucune caractéristique de l'ancien objet, mais les fils de liaison demeurent connectés à celui-ci.

Paramétrage des raccourcis clavier des commandes

Vous pouvez affecter des raccourcis clavier à des commandes de manière à ce que les utilisateurs puissent naviguer sur la face-avant sans souris. Effectuer un clic droit sur la commande puis sélectionnez **Avancé» Raccourci clavier** pour ouvrir la boîte de dialogue **Raccourci clavier**.

Lorsqu'un utilisateur entre un raccourci clavier alors que le VI est en cours d'exécution, ceci appelle la commande correspondante. Si cette commande est une commande texte, LabVIEW met le texte en surbrillance pour que vous puissiez l'éditer. Si cette commande est un booléen, appuyez sur la barre d'espace ou appuyez sur la touche <Entrée> pour changer sa valeur.

L'élément de menu local **Avancé»Raccourci clavier** est grisé pour les indicateurs car vous ne pouvez pas entrer de données dans un indicateur.

Contrôler le comportement d'un bouton avec le raccourci clavier

Vous pouvez associer des touches de fonction à divers boutons contrôlant le comportement d'une face-avant. Vous pouvez définir un bouton dans un VI pour qu'il se comporte comme une boîte de dialogue de sorte qu'appuyer sur la touche <Entrée> revienne à cliquer sur le bouton par défaut.

(Macintosh et Sun) Appuyer sur la touche <Return> revient à cliquer sur le bouton par défaut.

Si vous associez la touche <Entrée> ou la touche <Return> à un bouton de boîte de dialogue, LabVIEW l'entoure automatiquement d'un trait plus épais.

Si vous associez la touche <Entrée> ou la touche <Return> à une commande, aucune commande de chaîne de caractères de cette face-avant ne pourra recevoir de retour chariot. Par conséquent, toutes les chaînes de caractères de cette face-avant seront limitées à une seule ligne. Vous pouvez alors utiliser les barres de défilement pour naviguer sur des chaînes plus longues.

Si vous effectuez une boucle sur une commande booléenne et appuyez sur la touche <Entrée> ou <Return>, la commande booléenne change, même si une autre commande utilise la touche <Entrée> ou la touche <Return> en tant que raccourci clavier. Le raccourci clavier <Entrée> ou <Return> attribué n'est appliqué que lorsqu'aucune commande booléenne n'est sélectionnée.

Paramétrage de l'ordre de défilement des objets de la face-avant

Vous pouvez définir l'ordre de navigation des objets de la face-avant en sélectionnant **Édition»Définir l'ordre de défilement par tabulation.....** Vous pouvez alors utiliser la touche <Tab> pour naviguer parmi ces objets pendant l'exécution du VI.

Les commandes et indicateurs d'une face-avant ont un ordre, appelé ordre de la face-avant, qui n'a pas de rapport avec leur position sur la face-avant. La première commande ou le premier indicateur que vous créez sur la face-avant est l'élément 0, le deuxième est l'élément 1 et ainsi de suite. Si vous supprimez une commande ou un indicateur, l'ordre de la face-avant est ajusté automatiquement.

L'ordre de la face-avant détermine l'ordre de défilement lorsqu'un VI est en cours d'exécution. L'ordre de la face-avant détermine également l'ordre dans lequel les commandes et les indicateurs apparaissent dans les enregistrements des fichiers journaux que vous avez créés en enregistrant les données de la face-avant. Reportez-vous à la section [Enregistrement des données de la face-avant](#) du chapitre 13, *E/S sur fichiers*, pour de plus amples informations sur l'enregistrement de données.

Pour éviter que des utilisateurs n'accèdent à une commande en utilisant la touche <Tab> alors que le VI est en cours d'exécution, cochez la case **Ignorer cette commande lors de l'utilisation de la touche Tabulation** de la boîte de dialogue **Raccourci clavier**.

Coloriage des objets

Vous pouvez changer la couleur de beaucoup d'objets mais pas de tous. Par exemple, les terminaux du diagramme des objets de la face-avant et les fils de liaison utilisent des couleurs spécifiques en fonction du type et de la représentation des données qu'ils transportent ; vous ne pouvez donc pas modifier leurs couleurs.

Utilisez l'outil Pinceau et cliquez avec le bouton droit sur un objet ou sur l'espace de travail pour ajouter ou changer la couleur des objets de la face-avant ou des zones de travail de la face-avant et du diagramme. Vous pouvez également changer les couleurs par défaut de la plupart des objets en sélectionnant **Outils»Options...** puis **Couleurs** dans le menu déroulant du haut.

Utilisation d'images importées

Vous pouvez importer des images provenant d'autres applications pour les utiliser en arrière-plan de la face-avant, en éléments des commandes menus déroulants d'images et avec certaines parties d'autres commandes et indicateurs. Reportez-vous à la note d'application *LabVIEW Custom Controls, Indicators, and Type Definitions* pour obtenir de plus amples informations sur l'utilisation d'images dans les commandes.

Pour importer une image, copiez-la dans le presse-papiers et collez-la sur la face-avant. Vous pouvez aussi sélectionner **Édition»Importer un fichier image....**

Reportez-vous au fichier `examples\general\controls\custom.llb` pour obtenir des exemples de commandes contenant des graphiques importés.

Groupement et verrouillage des objets

Vous pouvez utiliser l'outil Flèche pour sélectionner les objets de la face-avant que vous voulez regrouper et verrouiller. Cliquez sur le bouton **Réorganiser** de la barre d'outils et sélectionnez **Grouper** ou **Verrouiller** dans le menu déroulant. Les objets groupés conservent leur taille et leur position les uns par rapport aux autres quand vous utilisez l'outil Flèche pour les déplacer ou modifier leur taille. Les objets verrouillés conservent leur position sur la face-avant et vous ne pouvez pas les supprimer tant que vous ne les avez pas déverrouillés. Vous pouvez paramétrer des objets pour qu'ils soient simultanément groupés et verrouillés. Les outils autres que l'outil Flèche fonctionnent normalement avec les objets groupés ou verrouillés.

Redimensionnement des objets

Vous pouvez changer la taille de la plupart des objets de la face-avant. Lorsque vous placez l'outil Flèche sur un objet redimensionnable, des poignées de redimensionnement apparaissent au niveau des coins de l'objet rectangulaire et des cercles de redimensionnement apparaissent sur un objet circulaire. Lorsque vous redimensionnez un objet, la taille de la police est conservée. Redimensionner un groupe d'objets redimensionne tous les objets de ce groupe.

La taille de certains objets, tels que les commandes et indicateurs numériques, ne peut être modifiée qu'horizontalement ou verticalement. D'autres conservent leurs proportions lorsque vous les redimensionnez. Le curseur de positionnement conserve son apparence, mais le bord en pointillés qui entoure l'objet ne se déplace que dans une direction.

Vous pouvez restreindre manuellement la direction de l'agrandissement lorsque vous redimensionnez un objet. Pour restreindre l'agrandissement, verticalement ou horizontalement, ou encore pour conserver les mêmes proportions, appuyez sur la touche <Maj> lorsque vous cliquez sur l'objet et le faites glisser. Pour redimensionner un objet autour de son point central, appuyez sur <Ctrl-Maj> et cliquez sur les curseurs de redimensionnement.

(Macintosh) Appuyez sur les touches <Option-Shift>. **(Sun)** Appuyez sur les touches <Meta-Shift>. **(Linux)** Appuyez sur les touches <Alt-Shift>.

Mise à l'échelle des objets de la face-avant

Vous pouvez paramétrer l'échelle des objets de la face-avant ou les redimensionner automatiquement par rapport à la taille de la fenêtre, lorsque vous redimensionnez la fenêtre de la face-avant. Vous pouvez mettre un objet de la face-avant à l'échelle ou vous pouvez mettre tous les objets de la face-avant à l'échelle. Cependant, vous ne pouvez pas mettre plusieurs objets distincts sur la face-avant à l'échelle, sauf si vous les mettez tous à la même échelle, ou si vous les groupez tous d'abord. Pour mettre un objet à l'échelle, sélectionnez celui-ci et sélectionnez **Édition»Mettre l'objet à l'échelle avec la face-avant**.

Si vous ne mettez qu'un seul objet de la face-avant à l'échelle, celui-ci se redimensionne automatiquement par rapport à toutes les modifications de la taille de la fenêtre de la face-avant ayant pu être effectuées. Les autres objets de la face-avant se repositionnent d'eux-mêmes pour demeurer cohérents avec leur position précédente sur la face-avant, mais ils ne se mettent pas à l'échelle pour correspondre à la nouvelle taille de la fenêtre sur la face-avant.

Dès que vous avez désigné un objet unique de la face-avant pour qu'il se mette automatiquement à l'échelle, des lignes grises définissent le contour de plusieurs zones de la face-avant, tel qu'illustré par la figure 4-1. Ces zones définissent les positions des autres objets de la face-avant par rapport à l'objet que vous désirez mettre à l'échelle. Lorsque vous redimensionnez la fenêtre de la face-avant, l'objet que vous avez mis automatiquement à l'échelle, se redimensionne et se repositionne par rapport à sa position d'origine. Les lignes grises disparaissent dès l'exécution du VI.

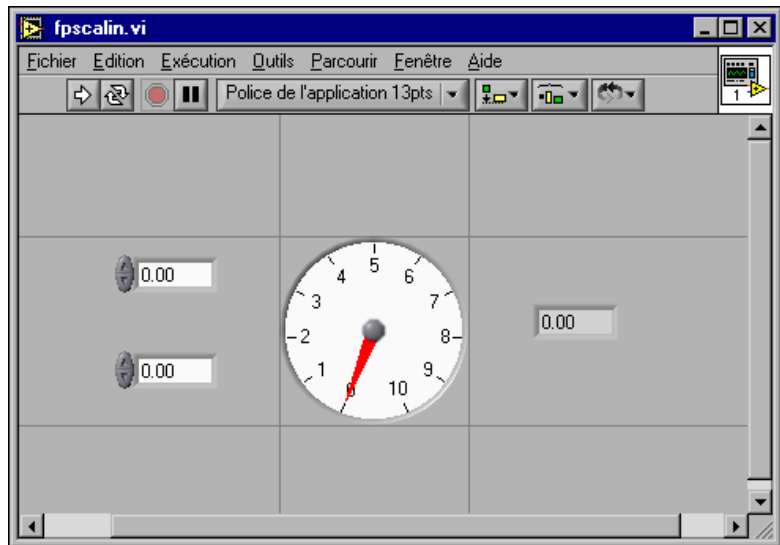


Figure 4-1. Face-avant avec objet mis à l'échelle

Lorsque LabVIEW met automatiquement des objets à l'échelle, il suit les mêmes conventions que celles du redimensionnement manuel des objets. Par exemple, certains objets ne peuvent être redimensionnés qu'horizontalement ou verticalement et la taille de la police est conservée lorsque vous redimensionnez un objet.

Si LabVIEW a mis un objet automatiquement à l'échelle, il est possible que celui-ci ne reprenne pas exactement sa taille d'origine quand vous redimensionnez la fenêtre conformément à sa position d'origine. Avant d'enregistrer le VI, sélectionnez **Édition**»**Annuler** pour restaurer les tailles d'origine de la fenêtre et des objets de la face-avant.

Vous pouvez mettre un tableau à l'échelle ou mettre les objets contenus dans un tableau à l'échelle. Si vous mettez un tableau à l'échelle, vous ajustez le nombre de lignes et de colonnes que vous pouvez afficher. Si vous mettez les objets d'un tableau à l'échelle, vous voyez toujours le même

nombre de lignes et de colonnes dans le tableau, malgré le fait que les tailles soient différentes.

Vous pouvez mettre à l'échelle soit un cluster, soit les objets contenus dans ce cluster. Si vous mettez les objets du cluster à l'échelle, le cluster s'ajustera également.

Ajout d'espace dans la face-avant sans redimensionner la fenêtre

Vous pouvez ajouter de l'espace dans la face-avant sans redimensionner la fenêtre. Pour augmenter l'espace entre des objets encombrant la fenêtre ou étroitement groupés, appuyez sur la touche <Ctrl> et utilisez l'outil Flèche pour cliquer sur la zone de travail de la face-avant. Tout en maintenant ces touches enfoncées, faites glisser l'outil jusqu'à l'obtention d'un rectangle de la taille désirée.

(Macintosh) Appuyez sur la touche <Option>. **(Sun)** Appuyez sur la touche <Meta>. **(Linux)** Appuyez sur la touche <Alt>.

Un rectangle entouré de pointillés définit l'endroit où cet espace va être inséré. Relâchez ces deux touches pour ajouter l'espace.

Commandes et indicateurs de la face-avant

Utilisez les commandes et indicateurs de la face-avant, situés sur la palette de **Commandes**, pour construire votre face-avant. Les commandes sont des boutons rotatifs, des boutons-poussoirs, des cadrans et autres commandes d'entrée. Les indicateurs sont des graphes, des LED et autres afficheurs. Les commandes simulent les périphériques d'entrée d'instruments et fournissent les données au diagramme du VI. Les indicateurs simulent les périphériques de sortie d'instruments et affichent les données que le diagramme acquiert ou génère.

Commandes et indicateurs 3D et classiques

De nombreux objets de la face-avant ont un aspect tridimensionnel et utilisent un nombre de couleurs élevé. Configurez votre carte vidéo pour afficher les couleurs avec une résolution d'au moins 16 bits afin d'obtenir l'aspect optimal des objets.

Aux objets 3D de la face-avant correspondent aussi des objets 2D utilisant un nombre de couleurs réduit. Utilisez les commandes et indicateurs 2D situés sur la palette **Commandes**»**Commandes classiques** afin de créer des VIs compatibles avec des affichages 256 et 16 couleurs.

Glissières, boutons rotatifs, cadrans et afficheurs numériques

Utilisez les commandes et indicateurs numériques, situés sur les palettes **Commandes»Numérique** et **Commandes»Commandes classiques»Numérique** pour simuler des glissières, des boutons rotatifs, des cadrans et des afficheurs numériques. La palette inclut également des boîtes de couleurs et une rampe de couleurs pour définir les valeurs des couleurs. Utilisez les commandes et indicateurs numériques pour entrer et afficher vos valeurs numériques.

Commandes et indicateurs à glissière(s)

Parmi les commandes et indicateurs à glissière(s), vous trouverez des glissières verticales et horizontales, un réservoir et un thermomètre. Changez la valeur de la commande ou de l'indicateur à glissière(s) en utilisant l'outil Doigt pour déplacer la glissière, en cliquant sur un point de l'objet glissière ou en utilisant l'afficheur numérique optionnel. Si vous déplacez la glissière et que le VI est en cours d'exécution durant cette modification, la commande lui transmet les valeurs intermédiaires, en fonction de la fréquence à laquelle le VI lit la commande.

Les commandes et indicateurs à glissière(s) peuvent afficher plusieurs valeurs. Cliquez avec le bouton droit sur l'objet puis sélectionnez **Ajouter une glissière** dans le menu local pour ajouter d'autres glissières. Le type de données d'une commande avec plusieurs glissières est un cluster contenant chacune des valeurs numériques. Reportez-vous à la section [Clusters](#) du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur les clusters.

Commandes et indicateurs rotatifs

Parmi les commandes et indicateurs rotatifs, vous trouverez des boutons rotatifs, des cadrans, des jauges et des vumètres. Les objets rotatifs fonctionnent de manière similaire aux commandes et indicateurs à glissières. Changez la valeur de la commande ou de l'indicateur rotatif en déplaçant les aiguilles, en cliquant sur un point de l'objet rotatif ou en utilisant l'affichage numérique optionnel.

Les commandes et indicateurs rotatifs peuvent afficher plusieurs valeurs. Cliquez avec le bouton droit sur l'objet et sélectionnez **Ajouter une aiguille** dans le menu local pour ajouter de nouvelles aiguilles. Le type de données d'une commande avec plusieurs aiguilles est un cluster contenant chacune des valeurs numériques. Reportez-vous à la section [Clusters](#) du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur les clusters.

Commandes et indicateurs numériques

Les commandes et indicateurs numériques vous offrent la manière la plus simple d'entrer et d'afficher des données numériques. Vous pouvez redimensionner horizontalement les objets de la face-avant pour qu'ils puissent contenir plus de chiffres. Vous pouvez changer la valeur d'une commande ou d'un indicateur numérique en utilisant les méthodes suivantes :

- Utilisez l'outil Doigt ou l'outil Texte à l'intérieur de la fenêtre d'affichage numérique puis entrez les nombres à l'aide du clavier.
- Utilisez l'outil Doigt et cliquez sur les flèches d'incrémentement ou de décrémentation d'une commande numérique.
- Utilisez l'outil Doigt ou l'outil Texte pour placer le curseur à droite du chiffre que vous désirez modifier et appuyez sur la touche fléchée, vers le haut ou vers le bas, du clavier.

Boîtes de couleur

Une boîte de couleurs affiche une couleur correspondant à une valeur spécifiée. Par exemple, vous pouvez utiliser les boîtes de couleur pour indiquer diverses conditions, telles que les valeurs hors gamme. La valeur de la couleur est exprimée sous forme de nombre hexadécimal avec la forme RRVVBB. Les deux premiers chiffres contrôlent la valeur de la couleur “ rouge.” Les deux chiffres suivants contrôlent la valeur de la couleur “ verte.” Les deux derniers chiffres contrôlent la valeur de la couleur “ bleu.”

Définissez la couleur de la boîte de couleur en cliquant dessus avec l'outil Doigt ou l'outil Pinceau afin d'afficher le sélecteur de couleur.

Rampes de couleurs

Une rampe de couleurs utilise une couleur pour afficher sa valeur numérique. Une échelle de couleurs est constituée d'au moins deux repères arbitraires, dont chacun a une valeur numérique et une couleur d'affichage correspondante. Lorsque la valeur d'entrée change, la couleur affichée se transforme pour correspondre à cette valeur. Les rampes de couleurs sont utiles pour indiquer visuellement les gammes de données, telles qu'une gamme de mise en garde avertissant l'utilisateur lorsqu'une jauge atteint une valeur dangereuse. Vous pouvez par exemple utiliser une rampe de couleurs pour paramétrer l'échelle de couleur utilisée pour les graphes et

graphes déroulants d'intensité. Reportez-vous à la section *Graphes et graphes déroulants d'intensité* du chapitre 11, *Graphes et graphes déroulants*, pour obtenir de plus amples informations sur les graphiques et graphes déroulants d'intensité.

Faites un clic droit sur la rampe de couleurs et utilisez les options du menu local pour personnaliser l'apparence, la taille, les couleurs et le nombre de couleurs.

Vous pouvez également ajouter une rampe de couleurs à chaque bouton rotatif, cadran ou jauge de la face-avant. Les vumètres ont, par défaut, une rampe de couleurs visible.

Boutons, commutateurs et voyants

Utilisez les commandes et indicateurs booléens, situés sur les palettes **Commandes»Booléen** et **Commandes»Commandes classiques»Booléen** pour simuler des boutons, des commutateurs et des voyants. Utilisez les commandes et indicateurs booléens pour entrer et afficher des valeurs booléennes (VRAI/FAUX). Par exemple, si vous surveillez la température d'une expérience, vous pouvez placer un témoin lumineux de mise en garde booléen sur la face-avant pour indiquer une surchauffe par rapport à une température donnée.

Utilisez le menu local afin de personnaliser l'apparence des objets booléens et leur comportement lorsque vous cliquez dessus.

Affichages des zones de saisie de texte, étiquettes et chemins

Utilisez les commandes et indicateurs de type chaîne et chemin, situés sur les palettes **Commandes»Chaîne & chemin** et **Commandes»Commandes classiques»Chaîne & chemin**, pour simuler des zones de saisie de texte ou des étiquettes et pour entrer ou afficher l'emplacement d'un fichier ou d'un répertoire.

Commandes et indicateurs de type chaîne

Utilisez les outils Doigt et Texte pour saisir ou modifier du texte dans une commande chaîne de la face-avant. Par défaut, le texte nouveau ou modifié n'est pas transmis au diagramme tant que vous n'avez pas terminé la session d'édition. Vous terminez la session d'édition en cliquant ailleurs sur la face-avant, en passant à une autre fenêtre, en cliquant sur le bouton **Valider** sur la barre d'outils ou en appuyant sur la touche <Entrée> du pavé numérique. Le fait d'appuyer sur la touche <Entrée> du clavier provoque un retour chariot.

(Macintosh et Sun) Le fait d'appuyer sur la touche <Return> du clavier provoque un retour chariot.

Reportez-vous à la section *Chaînes sur la face-avant* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur la commande et l'indicateur de type chaîne.

Commandes et indicateurs de type chemin

Utilisez les commandes et indicateurs de type chemin pour entrer ou afficher l'emplacement d'un fichier ou d'un répertoire. Les commandes et indicateurs de type chemin fonctionnent comme les commandes et indicateurs de type chaîne, mais LabVIEW formate le chemin en utilisant la syntaxe standard pour la plate-forme que vous utilisez.

Chemins invalides

En cas d'échec d'une fonction renvoyant un chemin, la fonction renvoie une valeur de chemin invalide, `Pas un chemin`, dans l'indicateur. Utilisez la valeur `Pas un chemin` comme valeur par défaut pour une commande de type chemin, de manière à pouvoir détecter le moment où l'utilisateur ne fournit pas un chemin et afficher une boîte de dialogue de fichier contenant des options pour la sélection d'un chemin. Utilisez la fonction Boîte de dialogue de fichier pour afficher une boîte de dialogue de fichier.

Chemins vides

Un chemin vide dans une commande de type chemin apparaît sous forme de chaîne vide sous Windows et Macintosh et prend la forme d'une barre oblique (/) sous UNIX. Utilisez les chemins vides pour inviter l'utilisateur à spécifier un chemin. Lorsque vous câblez un chemin vide à une fonction d'entrée/sortie sur fichiers, le chemin vide fait référence à la liste des disques connectés à l'ordinateur.

(Macintosh) Le chemin vide fait référence aux volumes montés.

(UNIX) Le chemin vide fait référence au répertoire racine.

Commandes et indicateurs de type tableau et cluster

Utilisez les commandes et indicateurs de type tableau et cluster situés sur les palettes **Commandes»Tableau et Cluster** et **Commandes»Commandes classiques»Tableau et Cluster** pour créer des tableaux et des clusters d'autres commandes et indicateurs. Reportez-vous à la section *Groupement des données avec des tableaux et des clusters* du Chapitre 9,

Groupement des données au moyen de chaînes, de tableaux et de clusters, pour obtenir de plus amples informations sur les tableaux et les clusters.

Les palettes **Tableau et Cluster** contiennent également les commandes et indicateurs de cluster d'erreur standard ainsi que les commandes et indicateurs à onglet. Reportez-vous à la section *Clusters d'erreur* du chapitre 6, *Exécution et mise au point des VIs*, pour obtenir de plus amples informations sur les clusters d'erreur. Reportez-vous à la section *Manipulation de données variant* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur les commandes de type variant.

Commandes à onglets

La commande onglet permet de regrouper et d'empiler des commandes sur plusieurs pages, ce qui permet de réduire la taille de votre face-avant. Une commande onglet comprend des pages et des onglets. Placez des objets de la face-avant sur chaque page d'une commande onglet et utilisez l'onglet comme sélecteur pour l'affichage des différentes pages. Vous pouvez placer un nombre illimité d'objets de la face-avant sur une commande onglet.

Les commandes onglet sont utiles lorsque vous avez de nombreux objets de la face-avant utilisés ensemble ou au cours d'une phase spécifique d'opération. Par exemple, vous pouvez avoir un VI qui nécessite une configuration préalable de plusieurs paramètres de la part de l'utilisateur avant le début d'un test. Ensuite, ce VI autorise l'utilisateur à modifier des aspects du test au cours de sa progression. Enfin, il autorise l'utilisateur à afficher et stocker uniquement les données pertinentes.

Sur le diagramme, la commande onglet est une commande de type énumération à la différence de commandes regroupées dans un cluster. Les terminaux des commandes et indicateurs placés sur la commande onglet apparaissent comme tout autre terminal de diagramme. Reportez-vous à la section *Commandes de type énumération* de ce chapitre pour obtenir de plus amples informations sur les commandes de type énumération.

Listes déroulantes

Utilisez les commandes de type liste déroulante, situées sur les palettes **Commandes»Liste & Table** et **Commandes»Commandes classiques»Liste & Table**, pour fournir aux utilisateurs une liste d'éléments parmi lesquels ils peuvent effectuer une sélection. Vous pouvez configurer des listes déroulantes de façon à ce qu'elles acceptent une seule ou plusieurs

sélections. Utilisez la liste multi-colonnes pour afficher davantage d'informations sur chaque élément, telles que la taille de l'élément et la date à laquelle il a été créé.

Utilisez le Nœud de propriété de la commande liste déroulante pour modifier les éléments de la liste et pour rassembler des informations sur les éléments de la liste des manières suivantes :

- Définir les chaînes des éléments.
- Ajouter un symbole à côté d'un élément de la liste, comme dans la boîte de dialogue **Enregistrer**, où les répertoires et les fichiers ont des symboles différents (boîte de dialogue non native uniquement).
- Désactiver certains éléments de la liste.
- Insérer des lignes séparatrices entre des éléments d'une liste.
- Détecter les éléments actuellement sélectionnés en lisant la valeur de la commande.
- Détecter les éléments sur lesquels l'utilisateur a éventuellement double-cliqué.

Reportez-vous à la section *Nœuds de propriété* du chapitre 16, *Contrôle des Vis par programmation*, pour obtenir de plus amples informations sur les nœuds de propriété.

Les listes déroulantes prennent en charge la saisie semi-automatique, ce qui signifie que vous tapez seulement les premiers caractères et LabVIEW recherche tous les éléments correspondants dans la liste déroulante. Appuyez sur la touche <Tab> pour vous déplacer à l'élément correspondant suivant. Appuyez sur la combinaison de touches <Maj-Tab> pour vous déplacer à l'élément correspondant précédent.

Les listes déroulantes possèdent automatiquement des barres de défilement, mais les barres de défilement sont activées uniquement lorsque la liste déroulante comprend plus d'informations qu'elle ne peut en afficher.

Commandes et indicateurs de type énumération et menu déroulant

Utilisez les commandes et indicateurs de type énumération et menu déroulant, situés sur les palettes **Commandes»Menu déroulant & Énum** et **Commandes»Commandes classiques»Menu déroulant & Énum**, pour créer une liste de chaînes que vous pouvez parcourir.

Commandes de type menu déroulant

Les commandes de type menu déroulant sont des objets numériques qui associent des valeurs numériques à des chaînes ou des images. Elles apparaissent sous forme de menus déroulants que les utilisateurs peuvent parcourir pour effectuer leurs sélections.

Les commandes de type menu déroulant sont utiles pour la sélection d'éléments mutuellement exclusifs, tels que des modes de déclenchement. Par exemple, utilisez une commande menu déroulant pour que les utilisateurs choisissent entre un déclenchement continu, unique et externe.

L'ordre des éléments dans la commande menu déroulant repose sur l'ordre dans lequel vous avez entré les éléments. Chaque élément possède une valeur numérique allant de 0 à $n-1$, où n est le nombre d'éléments. La commande menu déroulant affiche le dernier élément si l'utilisateur entre une valeur supérieure ou égale à $n-1$ et le premier élément si l'utilisateur entre une valeur inférieure ou égale à zéro.

Les commandes de type menu déroulant prennent en charge la saisie semi-automatique et les menus déroulants peuvent avoir des barres de défilement.

Commandes de type énumération

Utilisez les commandes de type énumération pour proposer aux utilisateurs une liste d'actions parmi lesquelles ils peuvent effectuer leur choix. Une commande de type énumération est similaire à une commande de menu déroulant de chaînes. Toutefois, avec une commande de type énumération, la valeur est une chaîne au lieu d'être un nombre comme pour les commandes de menu déroulant. Par exemple, vous pouvez utiliser une commande de type énumération pour sélectionner les conditions d'une structure Condition. Reportez-vous à la section *Structures Condition* du chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour obtenir de plus amples informations sur les Structures Condition.

Le type de données de la commande de type énumération est un octet non signé, un mot non signé ou un mot long non signé. Effectuez un clic droit sur une commande de type énumération et sélectionnez **Représentation** dans le menu local pour modifier le type de données de la commande.

Propriétés avancées des commandes et indicateurs de type énumération

Toutes les fonctions arithmétiques à l'exception de Incrémenter et Décrémenter traitent la commande de type énumération de la même façon qu'un nombre non signé. “ Incrémenter ” incrémente de la dernière valeur énumérée à la première et “ Décrémenter ” décrémente de la première valeur énumérée à la dernière. Lorsqu'on utilise un entier non signé dans une commande ou un indicateur de type énumération, les nombres négatifs sont modifiés de manière à être égaux à la première valeur énumérée et les nombres positifs hors gamme sont modifiés de manière à être égaux à la dernière valeur énumérée. Les entiers non signés hors gamme sont toujours modifiés de manière à être égaux à la dernière valeur énumérée.

Si vous connectez une valeur à virgule flottante à un indicateur de type énumération, le nombre est converti à l'élément d'énumération le plus proche, qui est l'une des valeurs dans l'indicateur de type énumération. LabVIEW manipule les nombres hors gamme, comme décrit auparavant. Si vous connectez une commande de type énumération à une valeur numérique, la valeur est l'indice du type énumération. Pour câbler une commande de type énumération à un indicateur de type énumération, les éléments de l'énumération doivent correspondre. L'indicateur peut avoir des éléments supplémentaires au-delà des éléments contenus dans la commande.

Commandes et indicateurs de nom d'E/S

Utilisez les commandes et indicateurs de nom d'E/S, situés sur les palettes **Commandes»E/S** ou **Commandes»Commandes classiques»E/S**, pour transmettre des noms de voie DAQ, des noms de ressource VISA et des noms logiques IVI pour configurer les VIs d'E/S afin de communiquer avec un instrument ou un périphérique d'acquisition de données.

Les constantes de nom d'E/S sont situées sur les palettes **Fonctions»E/S d'instruments** et **Fonctions»Acquisition de données**.

(Windows) Utilisez Measurement & Automation Explorer, disponible dans le menu **Outils**, pour configurer les noms de voie DAQ, les noms de ressource VISA et les noms logiques IVI.

(Macintosh) Utilisez l'utilitaire de configuration NI-DAQ, disponible dans le menu **Outils**, pour configurer le matériel DAQ National Instruments. Utilisez l'Assistant de voie DAQ, disponible dans le menu **Outils**, pour configurer les noms de voie DAQ.

(Macintosh et UNIX) Utilisez les utilitaires de configuration de votre instrument pour configurer les noms de ressource VISA et les noms logiquesIVI. Reportez-vous à la documentation fournie avec votre instrument pour obtenir de plus amples informations sur les utilitaires de configuration.

La commande IMAQ session est un identificateur unique qui représente la connexion au matériel.

Commande waveform

Utilisez la commande waveform située sur les palettes **Commandes»E/S** ou **Commandes»Commandes classiques»E/S** pour manipuler individuellement des éléments de données d'une waveform. Reportez-vous à la section *Type de données waveform* du chapitre 11, *Graphes et graphes déroulants* pour de plus amples informations à propos du type de données waveform.

Références à des objets ou à des applications

Utilisez les commandes et indicateurs de numéro de référence, situés sur les palettes **Commandes»Refnum** et **Commandes»Commandes classiques»Refnum**, pour travailler avec des fichiers, répertoires, périphériques et connexions réseau. Utilisez les commandes de type refnum pour transférer les informations des objets de la face-avant aux sous-VIs. Reportez-vous à la section *Contrôle des objets de la face-avant* du chapitre 16, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur les commandes refnum.

Un numéro de référence (ou refnum) est une référence unique d'un objet, tel qu'un fichier, un périphérique ou une connexion réseau. Lorsque vous ouvrez un fichier, un périphérique ou une connexion réseau, LabVIEW crée un refnum associé à ce fichier, ce périphérique ou cette connexion réseau. Toutes les opérations que vous effectuez sur les fichiers, périphériques ou connexions réseau ouverts utilisent des refnums pour identifier chaque objet. Utilisez une commande ou un indicateur de type refnum pour transmettre un refnum dans ou en dehors d'un VI. Par exemple, utilisez une commande ou un indicateur refnum pour modifier le contenu d'un fichier référencé par un refnum sans fermer et rouvrir le fichier.

Un refnum étant un pointeur temporaire sur un objet, il est valide uniquement pour la période pendant laquelle l'objet est ouvert. Si vous fermez l'objet, LabVIEW dissocie le refnum de l'objet et le refnum devient obsolète. Si vous ouvrez à nouveau l'objet, LabVIEW crée un nouveau refnum qui est différent du premier.

LabVIEW garde en mémoire les informations associées à chaque refnum, telles que l'emplacement actuel (pour y lire ou y écrire) et le degré d'accès de l'utilisateur. Vous pouvez ainsi effectuer des opérations simultanées mais indépendantes sur un seul objet. Si un VI ouvre un objet plusieurs fois de suite, chaque opération d'ouverture retourne un refnum différent.

Commandes de boîte de dialogue

Utilisez les commandes de boîte de dialogue, situées sur la palette **Commandes»Commandes (dialogues)**, dans les boîtes de dialogue que vous créez. Les commandes et indicateurs de boîte de dialogue sont conçus spécialement pour une utilisation dans des boîtes de dialogue et comprennent des commandes de menu déroulant, des boutons, des onglets, des cases à cocher et des boutons radio. Seule l'apparence diffère entre ces commandes et celles de la face-avant. Ces commandes apparaissent avec les couleurs que vous avez définies pour votre bureau.

Étant donné que l'apparence des commandes de boîte de dialogue change en fonction de la plate-forme sur laquelle vous exécutez le VI, l'apparence des commandes dans les VIs que vous créez est compatible avec toutes les plates-formes de LabVIEW. Lorsque vous exécutez le VI sur une autre plate-forme, les commandes de boîte de dialogue adaptent leur couleur et leur apparence afin de correspondre aux commandes de boîte de dialogue standards de la plate-forme en question.

Sélectionnez **Fichier»Propriétés du VI...** et choisissez **Apparence de la fenêtre** dans le menu déroulant **Catégorie** pour masquer la barre de menus et les barres de défilement et pour créer des VIs qui imitent l'apparence et le comportement des boîtes de dialogue standards de chaque plate-forme. Reportez-vous à la section [Configuration de l'apparence et du comportement des VIs](#) du chapitre 15, [Personnalisation des VIs](#), pour obtenir de plus amples informations sur la configuration de l'apparence et du comportement des VIs.

Étiquetage

Utilisez des étiquettes pour identifier les objets de la face-avant et du diagramme.

LabVIEW possède deux sortes d'étiquettes : étiquettes dépendantes et étiquettes libres. Les étiquettes dépendantes appartiennent et se déplacent avec un objet particulier et annotent uniquement cet objet. Vous pouvez déplacer seulement une étiquette dépendante, mais lorsque vous déplacez l'objet propriétaire de l'étiquette, cette dernière évolue avec l'objet. Vous

pouvez masquer des étiquettes dépendantes, mais vous ne pouvez pas les copier ou les effacer indépendamment de leurs propriétaires. Vous pouvez aussi afficher une étiquette d'unités pour les commandes numériques en sélectionnant **Éléments visibles»Étiquette** dans le menu local. Reportez-vous à la section *Unités numériques et vérification de type stricte* du chapitre 5, *Construction du diagramme*, pour de plus amples informations à propos des unités numériques.

Les étiquettes libres ne sont attachées à aucun objet et vous pouvez les créer, déplacer, faire pivoter ou les supprimer indépendamment. Utilisez-les pour vos annotations sur les faces-avant et sur les diagrammes.

Utiliser l'outil Texte pour créer des étiquettes libres ou pour éditer l'un ou l'autre type d'étiquettes.

Sous-titres

Les objets de la face-avant peuvent aussi avoir des sous-titres. Cliquez avec le bouton droit sur l'objet et sélectionnez **Éléments visibles»Sous-titre** dans le menu local pour afficher le sous-titre. À la différence d'une étiquette, un sous-titre n'affecte pas le nom de l'objet et vous pouvez l'utiliser comme une étiquette d'objet encore plus descriptive. Le sous-titre n'apparaît pas sur le diagramme.

Caractéristiques du texte

LabVIEW utilise les polices déjà installées sur votre ordinateur. Utilisez le menu déroulant **Paramètres du texte** sur la barre d'outils afin de modifier les attributs du texte. Si vous sélectionnez des objets ou du texte avant d'effectuer un choix dans le menu déroulant **Paramètres du texte**, les modifications s'appliquent à tout ce que vous avez sélectionné. Si vous ne sélectionnez rien, les modifications s'appliquent à la police par défaut. Le changement de la police par défaut ne modifie pas celle des étiquettes existantes. Il affecte uniquement les étiquettes que vous créez à partir de cet instant.

Choisissez la **Sélection de la police par défaut** à partir du menu déroulant **Paramètres du texte** sur la face-avant afin d'appliquer des styles de polices spécifiques au texte que vous avez sélectionné. Si vous ne sélectionnez aucun texte, l'option **Paramètre par défaut de la face-avant** est cochée. Si vous choisissez **Paramètres du texte»Sélection de la police par défaut** à partir du diagramme sans sélectionner aucun objet, l'option **Paramètre par défaut du diagramme** est cochée. Vous pouvez définir des polices

différentes pour la face-avant et le diagramme. Par exemple vous pouvez utiliser une police de petite taille sur le diagramme et une autre, plus grande, sur la face-avant.

Le menu déroulant **Paramètres du texte** contient les polices intégrées suivantes :

- **Police de l'application** : police utilisée par défaut pour les palettes **Commandes** et **Fonctions** et pour le texte dans les nouvelles commandes
- **Police du système** : utilisée pour les menus
- **Police des boîtes de dialogue** : utilisée pour du texte dans les boîtes de dialogue

Quand vous transférez un VI qui contient une de ces polices intégrées vers une autre plate-forme, les polices correspondent autant que possible.

Le menu déroulant **Paramètres du texte** possède aussi les éléments de sous-menu **Taille**, **Style**, **Justifier** et **Couleur**.

Les polices choisies le sont à partir de n'importe lequel de ces sous-menus et s'appliquent aux objets que vous avez sélectionnés. Par exemple, si vous choisissez une nouvelle police alors que sont déjà sélectionnés un bouton rotatif et un graphique, les étiquettes, les échelles et les afficheurs numériques seront tous modifiés avec la nouvelle police.

LabVIEW conserve autant d'attributs de polices que possible quand vous effectuez une modification. Par exemple, si vous changez plusieurs objets avec la police Courier, ces derniers conservent si possible leur taille et leur style. Quand vous utilisez la boîte de dialogue **Paramètres du texte**, LabVIEW modifie les objets que vous sélectionnez avec les caractéristiques de texte que vous avez retenues. Si vous sélectionnez une des polices intégrées ou la police actuelle, LabVIEW modifie les objets sélectionnés selon la police et la taille associées à cette police.

Quand vous travaillez avec des objets qui possèdent plusieurs éléments de texte, comme des diapositives, les modifications de police que vous apportez affectent les objets et le texte que vous avez actuellement sélectionnés. Par exemple, si vous sélectionnez la glissière en entier et ensuite **Style»Gras** dans le menu déroulant **Paramètres du texte**, l'échelle, l'afficheur numérique et l'étiquette seront modifiés avec une police Gras. Si vous sélectionnez seulement l'étiquette et ensuite **Gras**, seule cette dernière sera modifiée avec une police Gras. Si vous sélectionnez du texte à partir d'un repère de l'échelle et ensuite **Gras**, tous les marqueurs seront modifiés avec une police Gras.

Conception des interfaces utilisateur

Si un VI est utilisé en tant qu'interface utilisateur ou boîte de dialogue, l'apparence et la disposition de la face-avant sont importantes. Concevez la face-avant de sorte que les utilisateurs puissent facilement identifier les actions à accomplir. Vous pouvez concevoir des faces-avant ressemblant à des instruments ou à d'autres périphériques.

Reportez-vous au manuel *LabVIEW Development Guidelines* pour obtenir plus d'informations sur la conception d'interfaces utilisateur.

Reportez-vous à la section *Structures Condition, Séquence et Événement* du chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour obtenir des informations sur l'utilisation d'événements afin d'améliorer la fonctionnalité des interfaces utilisateur.

Utilisation des commandes et indicateurs de la face-avant

Les commandes et les indicateurs sont les éléments principaux de la face-avant. Au cours de la conception de la face-avant, prenez en considération la logique d'interaction des utilisateurs avec les groupes de commandes et d'indicateurs du VI. Si plusieurs commandes sont apparentées, ajoutez un cadre décoratif autour de chacune d'entre elles ou incluez-les dans un cluster. Utilisez les décorations situées sur les palettes **Commandes»Décorations** et **Commandes»Commandes classiques»Décorations** pour grouper ou séparer des objets sur une face-avant à l'aide de cadres, lignes ou flèches. Ces objets sont à utiliser dans un but de décoration et non pas pour afficher des données.

Ne positionnez pas des objets de la face-avant trop près les uns des autres. Essayez de laisser un certain espace libre pour rendre la face-avant plus facile à lire. Cet espace libre empêchera les utilisateurs de cliquer par inadvertance sur la commande ou le bouton inadéquat.

Attribuez des noms explicites aux boutons et utilisez une terminologie courante. Utilisez des noms tels que Démarrer, Arrêter et Enregistrer plutôt que OK. Des noms explicites facilitent l'utilisation du VI.

Utilisez les polices et les couleurs par défaut de LabVIEW. LabVIEW remplace les polices intégrées par des familles de polices similaires sur des plates-formes différentes. Si vous sélectionnez une police différente, LabVIEW assure le remplacement par l'équivalence la plus proche si la police n'est pas disponible sur un ordinateur. LabVIEW gère les couleurs de la même manière que les polices de caractères. Si une couleur n'est pas

disponible sur un ordinateur, LabVIEW la remplace par l'équivalence la plus proche. Vous pouvez également utiliser les couleurs système pour adapter l'apparence d'une face-avant aux couleurs système de tout ordinateur exécutant le VI.

Évitez de positionner des objets par-dessus d'autres objets. Une étiquette ou tout autre objet recouvrant, même partiellement, une commande ou un indicateur ralentit les mises à jour de l'écran et peut provoquer le scintillement de cette commande ou indicateur.

Conception des boîtes de dialogue

Si un VI contient des boîtes de dialogue consécutives qui apparaissent sur le même emplacement de l'écran, organisez-les de façon à ce que les boutons de la première boîte de dialogue ne soient pas directement alignés avec ceux de la boîte de dialogue suivante. Les utilisateurs risqueraient de double-cliquer sur un bouton de la première boîte de dialogue sans se rendre compte qu'ils ont cliqué sur un bouton de la boîte de dialogue suivante. Reportez-vous à la section [Commandes de boîte de dialogue](#) de ce chapitre pour obtenir de plus amples informations à propos des commandes des boîtes de dialogue.

Sélection de la taille de l'écran

Quand vous concevez un VI, tenez compte des différentes résolutions d'écran des ordinateurs sur lesquels la face-avant s'affichera. Sélectionnez **Fichier»Propriétés du VI...**, sélectionnez **Taille de la fenêtre** dans le menu déroulant **Catégorie** et cochez la case **Conserver les proportions de la fenêtre pour différentes résolutions d'écran** afin de conserver les proportions de la fenêtre de la face-avant par rapport à la résolution de l'écran.

Construction du diagramme

Après avoir construit la face-avant, vous devez ajouter le code en utilisant les représentations graphiques des fonctions pour contrôler les objets de la face-avant. Le diagramme contient ce code source graphique.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations à propos de la conception et de la configuration du diagramme.

Relations entre les objets de la face-avant et les terminaux du diagramme

Les objets de la face-avant apparaissent en tant que terminaux sur le diagramme. Double-cliquez sur un terminal du diagramme pour mettre en surbrillance la commande ou l'indicateur correspondant sur la face-avant.

Les terminaux correspondent à des ports d'entrée et de sortie qui échangent des informations entre la face-avant et le diagramme. Les données que vous entrez sur les commandes de la face-avant entrent dans le diagramme par le biais des terminaux de commandes. Quand le VI termine son exécution, les données de sortie progressent vers les terminaux de l'indicateur, d'où elles quittent le diagramme, pour entrer à nouveau sur la face-avant et y apparaître dans les indicateurs.

Objets du diagramme

Les objets sur le diagramme se composent de terminaux, de nœuds et de fonctions. Vous pouvez construire des diagrammes en connectant les objets au moyen de fils de liaison.

Terminaux du diagramme



Les terminaux du diagramme représentent le type de données d'un indicateur ou d'une commande de la face-avant. Par exemple, un terminal DBL, tel que celui affiché à gauche, représente une commande ou un indicateur numérique à virgule flottante double précision.

Un terminal correspond à n'importe quel point, autre qu'un fil de liaison, auquel vous pouvez attacher un fil de liaison. LabVIEW possède des terminaux pour commandes et indicateurs, des terminaux de nœuds, des constantes et des terminaux spécialisés sur des structures, tels que des terminaux d'entrée et de sortie sur la boîte de calcul. Vous utilisez les fils de liaison pour connecter les terminaux et transmettre les données vers d'autres terminaux. Cliquez avec le bouton droit de la souris sur un objet du diagramme et sélectionnez **Éléments visibles»Terminaux** à partir du menu local pour afficher les terminaux. Cliquez avec le bouton droit de la souris sur l'objet et sélectionnez à nouveau **Éléments visibles»Terminaux** afin de masquer les terminaux. Cet élément de menu local n'est pas disponible pour les fonctions extensibles.

Type de données des indicateurs et des commandes

La table 5-1 présente les symboles pour différents types de terminaux d'indicateurs et de commandes. La couleur et le symbole de chaque terminal indiquent le type de données de la commande ou de l'indicateur. Les terminaux des commandes possèdent un cadre plus épais que celui des terminaux des indicateurs. De plus, une flèche apparaît soit sur la droite si le terminal appartient à une commande, soit sur la gauche si le terminal appartient à un indicateur.

Tableau 5-1. Terminaux des commandes et des indicateurs

Commande	Indicateur	Type de données	Couleur
		Numérique à virgule flottante simple précision	Orange
		Numérique à virgule flottante double précision	Orange
		Numérique à virgule flottante précision étendue	Orange
		Numérique complexe à virgule flottante simple précision	Orange
		Numérique complexe à virgule flottante double précision	Orange

Tableau 5-1. Terminaux des commandes et des indicateurs (Suite)









































Commande	Indicateur	Type de données	Couleur
		Numérique complexe à virgule flottante précision étendue	Orange
		Numérique entier 8 bits signé	Bleu
		Numérique entier 16 bits signé	Bleu
		Numérique entier 32 bits signé	Bleu
		Numérique entier 8 bits non signé	Bleu
		Numérique entier 16 bits non signé	Bleu
		Numérique entier 32 bits non signé	Bleu
		Type énumération	Bleu
		Booléen	Vert
		Chaîne	Rose
		Tableau : met entre crochets droits le type de données de ses éléments et prend la couleur de ce type de données.	Variable
 	 	Cluster : met entre crochets plusieurs types de données. Les types de données de cluster sont marrons si les éléments du cluster sont du même type ou roses si les éléments du cluster sont de types différents.	Marron ou rose
		Chemin	Turquoise
		Waveform : cluster d'éléments qui comporte les données, temps de départ et Δt d'un signal. Reportez-vous à la section <i>Type de données waveform</i> du chapitre 11, <i>Graphes et graphes déroulants</i> pour de plus amples informations à propos du type de données waveform.	Marron
		Numéro de référence (refnum)	Turquoise
		Variant : inclut le nom de la commande ou de l'indicateur, les informations du type de données et les données elles-mêmes. Reportez-vous à la section <i>Manipulation de données variant</i> de ce chapitre pour obtenir de plus amples informations à propos du type de données variant.	Violet

Tableau 5-1. Terminaux des commandes et des indicateurs (Suite)

Commande	Indicateur	Type de données	Couleur
		Polymorphe : indique qu'un VI ou une fonction accepte plusieurs sortes de types de données. Reportez-vous à la section <i>Vis et fonctions polymorphes</i> de ce chapitre pour obtenir de plus amples informations à propos des VIs et des fonctions polymorphes.	Violet
		Nom d'E/S : transmet les noms des voies DAQ, des ressources VISA et les noms logiques IVI que vous configurez en VIs d'E/S pour communiquer avec un instrument ou un périphérique DAQ. Reportez-vous à la section <i>Commandes et indicateurs de nom d'E/S</i> du chapitre 4, <i>Construction de la face-avant</i> pour de plus amples informations à propos du type de données des noms d'E/S.	Violet
		Image : affiche les images pouvant contenir des lignes, des cercles, du texte et d'autres types de formes graphiques. Reportez-vous à la section <i>Utilisation de l'indicateur image</i> du chapitre 12, <i>Vis Graphisme et son</i> pour de plus amples informations à propos du type de données images.	Bleu

De nombreux types de données possèdent un jeu de fonctions correspondant permettant de manipuler les données. Reportez-vous à la section *Sommaire des fonctions* de ce chapitre pour obtenir de plus amples informations à propos des fonctions à utiliser avec chaque type de données.

Constantes

Les constantes sont des terminaux sur le diagramme qui lui fournissent des données fixes. Les constantes universelles correspondent à des constantes avec des valeurs fixes, telles que pi (π) et l'infini (∞). Les constantes définies par l'utilisateur correspondent à des constantes que vous définissez et éditez avant d'exécuter le VI.

Étiquetez une constante en faisant un clic droit sur la constante et en sélectionnant **Éléments visibles**»**Étiquette** dans le menu local. Les constantes universelles possèdent des étiquettes prédéterminées que vous pouvez éditer en utilisant l'outil Texte.

La plupart des constantes se trouvent en haut ou en bas de leurs palettes.

Constantes universelles

Utilisez les constantes universelles pour les calculs mathématiques et le formatage des chaînes ou des chemins. LabVIEW inclut les types de constantes universelles suivants :

- **Constantes numériques universelles** : un ensemble de valeurs physiques et mathématiques de haute précision communément utilisées, telles que la base du logarithme naturel (e) et la vitesse de la lumière. Les constantes numériques universelles sont situées sur la palette **Fonctions»Numérique»Constantes numériques supplémentaires**.
- **Constantes chaînes universelles** : il s'agit d'un ensemble de caractères sous forme de chaîne communément utilisés et non affichables, tels que le retour à la ligne et le retour chariot. Les constantes chaînes universelles se trouvent sur la palette **Fonctions»Chaîne**.
- **Constantes de fichiers universelles** : il s'agit d'un ensemble de valeurs de chemins de fichier communément utilisées, telles que Pas un chemin, Pas un refnum et Répertoire par défaut. Les constantes fichiers universelles sont situées sur la palette **Fonctions»E/S sur fichiers»Constantes de fichiers**.

Les constantes définies par l'utilisateur

La palette **Fonctions** comprend des constantes classées selon les types suivants : booléen, numérique, menu déroulant, énumération, boîte de couleurs, liste déroulante, chaîne, tableau, cluster et constantes chemin.

Créez une constante définie par l'utilisateur en cliquant avec le bouton droit de la souris sur le terminal d'entrée d'un VI ou d'une fonction et en sélectionnant **Créer une constante** dans le menu local. Vous ne pouvez pas modifier la valeur des constantes définies par l'utilisateur quand le VI est en cours d'exécution.

Vous pouvez aussi créer une constante en faisant glisser une commande de la face-avant dans le diagramme. La constante qui en résulte contient la valeur de la commande de la face-avant au moment où vous la faites glisser vers le diagramme. La commande de la face-avant reste sur la face-avant. Changer la valeur de la commande n'affecte pas la valeur de la constante et vice versa.

Utilisez l'outil Doigt ou l'outil Texte pour cliquer sur la constante et changer sa valeur. Si la sélection automatique de l'outil est activée, double-cliquez sur la constante pour passer à l'outil Texte et éditer la valeur.

Nœuds du diagramme

Les nœuds sont des objets sur le diagramme qui ont des entrées et/ou des sorties et qui réalisent des opérations pendant l'exécution du VI. Ils sont similaires aux déclarations, opérateurs, fonctions et sous-programmes des langages de programmation textuels. LabVIEW inclut les types de nœuds suivants :

- **Fonctions** : éléments d'exécution intégrés, comparables à un opérateur, une fonction ou une déclaration. Reportez-vous à la section [Sommaire des fonctions](#) de ce chapitre pour obtenir de plus amples informations à propos des fonctions disponibles dans LabVIEW.
- **Sous-VIs** : VIs utilisés dans le diagramme d'un autre VI, comparables à des sous-programmes. Reportez-vous à la section [Sous-VIs](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour obtenir de plus amples informations à propos de l'utilisation des sous-VIs dans un diagramme.
- **Structures** : effectuent le traitement des éléments de commandes, tels que les structures Séquence, structures Condition, boucles For ou boucles While. Reportez-vous au chapitre 8, [Boucles For et While et Structures Condition, Séquence et Événement](#), pour obtenir de plus amples informations à propos de l'utilisation des structures.
- **Boîtes de calcul** : structures pouvant être redimensionnées pour entrer directement des équations dans un diagramme. Reportez-vous à la section [Boîtes de calcul](#) du chapitre 20, [Formules et équations](#), pour obtenir de plus amples informations à propos de l'utilisation des boîtes de calcul.
- **Nœuds de propriété** : définissent ou recherchent les propriétés d'une classe. Reportez-vous à la section [Nœuds de propriété](#) du chapitre 16, [Contrôle des VIs par programmation](#), pour obtenir de plus amples informations sur l'utilisation des nœuds de propriété.
- **Nœuds de méthode** : exécutent les méthodes d'une classe. Reportez-vous à la section [Nœuds de méthode](#) du chapitre 16, [Contrôle des VIs par programmation](#), pour obtenir de plus amples informations sur l'utilisation des nœuds de méthode.
- **Code Interface Nodes (CIN)** : appellent du code écrit dans des langages de programmation textuels. Reportez-vous à la section [Code Interface Node](#) du chapitre 19, [Appel de code provenant de langages de programmation textuels](#) pour obtenir de plus amples informations à propos de l'appel de code issu de langages de programmation textuels.



Après avoir construit la face-avant et le diagramme d'un VI, construisez le connecteur, représenté à gauche, afin de pouvoir utiliser le VI comme un sous-VI. Le connecteur s'apparente à un ensemble de terminaux qui correspondent aux commandes et aux indicateurs de ce VI et correspond à la liste de paramètres d'un appel de fonction dans les langages de programmation textuels. Le connecteur définit les entrées et sorties que vous pouvez connecter au VI que vous voulez utiliser comme sous-VI. Reportez-vous à la section [Configuration du connecteur](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour obtenir de plus amples informations sur la configuration des connecteurs.

Sommaire des fonctions

Les fonctions sont les éléments de fonctionnement essentiels de LabVIEW. Les icônes de fonction sur la palette de **Fonctions** ont un arrière-plan de couleur jaune clair et un avant-plan noir. Les fonctions n'ont pas de diagramme ni de face-avant, mais possèdent des connecteurs.

La palette de **Fonctions** contient aussi les VIs qui accompagnent LabVIEW. Utilisez ces VIs en tant que sous-VIs lorsque vous construisez des VIs d'acquisition de données, de commandes d'instruments, de communication ou autres. Reportez-vous à la section [Utilisation des VIs et fonctions intégrés](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour de plus amples informations à propos de l'utilisation des VIs intégrés.

Fonctions numériques

Utilisez les fonctions numériques, se trouvant sur la palette **Fonctions» Numérique** pour créer et réaliser des opérations arithmétiques, trigonométriques, logarithmiques et mathématiques complexes sur des nombres et pour convertir des nombres d'un type de données en un autre.

Fonctions booléennes

Utilisez les fonctions booléennes se trouvant sur la palette **Fonctions» Booléen** pour réaliser des opérations logiques sur des valeurs booléennes uniques ou sur des tableaux de valeurs booléennes, afin d'effectuer les tâches suivantes :

- Changer une valeur VRAI en une valeur FAUX et vice versa.
- Déterminer quelle valeur booléenne retourner si vous recevez deux valeurs booléennes ou plus.

- Convertir une valeur booléenne en un chiffre (1 ou 0).
- Utiliser un opérateur arithmétique sur deux valeurs booléennes ou davantage.

Fonctions chaîne

Utilisez les fonctions Chaîne, qui se trouvent sur la palette **Fonctions»chaîne** pour réaliser les tâches suivantes :

- Concaténer plusieurs chaînes.
- Extraire un sous-ensemble de chaînes d'une chaîne.
- Rechercher et remplacer des caractères ou des sous-ensembles de chaînes dans une chaîne.
- Convertir des données en chaînes.
- Formater une chaîne pour l'utiliser avec un traitement de texte ou un tableur.

Reportez-vous à la section *Chaînes* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations à propos de l'utilisation des fonctions chaîne.

Fonctions tableau

Utilisez les fonctions tableau, qui se trouvent sur la palette **Fonctions»Tableau** pour créer et manipuler des tableaux afin d'effectuer les tâches suivantes :

- Extraire des éléments de données d'un tableau.
- Ajouter des éléments de données individuels à un tableau.
- Diviser des tableaux.

Reportez-vous à la section *Tableaux* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations à propos de l'utilisation des fonctions tableau.

Fonctions cluster

Utilisez les fonctions Cluster, qui se trouvent sur la palette **Fonctions»Cluster** pour créer et manipuler des clusters afin d'effectuer les tâches suivantes :

- Extraire des éléments de données individuels d'un cluster.
- Ajouter des éléments de données individuels à un cluster.
- Réduire un cluster à ses éléments de données individuels.

Reportez-vous à la section *Clusters* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations à propos de l'utilisation des fonctions cluster.

Fonctions de comparaison

Utilisez les fonctions de comparaison situées dans la palette **Fonctions» Comparaison** pour comparer des valeurs booléennes, des chaînes de caractères, des nombres, des tableaux et des clusters.

Reportez-vous à l'annexe C, *Fonctions de comparaison*, pour obtenir de plus amples informations à propos de l'utilisation des fonctions de comparaison.

Fonctions Temps & Dialogue

Utiliser les fonctions Temps & Dialogue qui se trouvent sur la palette **Fonctions» Temps et Dialogue** afin de réaliser les tâches suivantes :

- Manipuler la vitesse d'exécution d'une opération.
- Récupérez les informations d'heure et de date à partir de l'horloge de votre ordinateur.
- Créer des boîtes de dialogue pour fournir des instructions aux utilisateurs.

Cette palette inclut aussi les VIs de gestion d'erreur. Reportez-vous à la section *Recherche et gestion d'erreur* du chapitre 6, *Exécution et mise au point des VIs* pour obtenir de plus amples informations à propos de l'utilisation des VIs Gestionnaire d'erreur.

Fonctions E/S sur fichiers

Utilisez les fonctions E/S sur fichiers, qui se trouvent sur la palette **Fonctions» E/S sur fichiers** pour effectuer les tâches suivantes :

- Ouvrir et fermer des fichiers.
- Lire et écrire sur des fichiers.
- Créer les répertoires et les fichiers que vous spécifiez dans la commande des chemins.
- Récupérer des informations de répertoires.
- Écrire des chaînes, des nombres, des tableaux et des clusters sur des fichiers.

La palette **E/S sur fichiers** comprend aussi des VIs qui réalisent des tâches communes d'E/S sur fichiers. Reportez-vous au chapitre 13, *E/S sur fichiers* pour obtenir de plus amples informations à propos de l'utilisation des VIs et des fonctions E/S sur fichiers.

Fonctions waveform

Utilisez les fonctions Waveform, qui se trouvent sur la palette **Fonctions»Waveform** pour réaliser les tâches suivantes :

- Construire des données de type waveform qui comprennent des valeurs du signal, des informations sur la voie et des informations de synchronisation.
- Extraire des éléments individuels de données waveform.
- Éditer des éléments individuels de données waveform.

Reportez-vous à la partie II, *DAQ Basics* du *LabVIEW Measurements Manual*, pour obtenir de plus amples informations à propos de la création et de l'utilisation des données waveform dans les VIs.

Fonctions de contrôle d'applications

Utilisez les fonctions de contrôle d'applications qui se trouvent sur la palette **Fonctions»Contrôle d'applications** pour contrôler par programmation les VIs et les applications LabVIEW sur votre ordinateur local ou sur un réseau. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation* pour obtenir de plus amples informations à propos de l'utilisation des fonctions de contrôle d'applications.

Fonctions avancées

Utilisez les fonctions avancées qui se trouvent sur la palette **Fonctions»Avancé** pour appeler le code en provenance des bibliothèques, telles que les bibliothèques de liens dynamiques (DLL), afin de manipuler les données LabVIEW et de les utiliser dans d'autres applications et pour appeler une portion de code provenant de langages de programmation textuels. Reportez-vous au manuel *Using External Code in LabVIEW* pour obtenir de plus amples informations à propos de l'utilisation des fonctions avancées.

Ajout de terminaux aux fonctions du diagramme

Vous pouvez modifier le nombre de terminaux pour certaines fonctions. Par exemple, pour construire un tableau avec dix éléments, vous devez ajouter dix terminaux.

Vous pouvez ajouter des terminaux aux fonctions extensibles en utilisant l'outil Flèche pour faire glisser le coin de la fonction. Vous pouvez aussi utiliser l'outil Flèche pour supprimer des terminaux des fonctions extensibles, mais vous ne pouvez pas supprimer un terminal qui est déjà câblé. Vous devez d'abord supprimer le fil de liaison existant pour supprimer le terminal.

Vous pouvez aussi ajouter ou supprimer des terminaux en cliquant avec le bouton droit de la souris sur l'un des terminaux de l'icône et en sélectionnant **Ajouter une entrée**, **Ajouter une sortie**, **Supprimer une entrée** ou **Supprimer une sortie à partir du menu local**. Vous pouvez ajouter des terminaux pour des commandes d'entrée, de sortie ou de type refnum, selon la fonction. Les rubriques **Ajouter une entrée** et **Ajouter une sortie** du menu local ajoutent immédiatement un terminal à la suite de celui sur lequel vous avez cliqué avec le bouton droit de la souris. Les rubriques **Supprimer une entrée** et **Supprimer une sortie** du menu local suppriment le terminal sur lequel vous avez cliqué avec le bouton droit de la souris. Si vous utilisez le menu local pour supprimer un terminal câblé, LabVIEW supprime le terminal et déconnecte le fil de liaison.

Utilisation des fils de liaison pour relier les objets du diagramme

Vous transférez les données entre les objets du diagramme au moyen des fils de liaison. Chaque fil de liaison a une source de données unique, mais vous pouvez le câbler à de nombreux VIs et fonctions qui lisent les données. Les fils de liaison ont des couleurs, des styles et des épaisseurs différents en fonction de leurs types de données. Un fil de liaison brisé apparaît comme une ligne en pointillé. Reportez-vous à la *Carte de référence rapide LabVIEW* pour obtenir de plus amples informations à propos des types de données des fils de liaison.

Des fils de liaison tronqués apparaissent aux extrémités des terminaux non câblés lorsque vous déplacez l'outil Bobine sur un VI ou un nœud de fonction. Ils indiquent le type de données de chaque terminal. Une info-bulle apparaît, répertoriant le nom du terminal. Après avoir câblé

un terminal, le fil tronqué de ce terminal n'apparaît plus lorsque vous déplacez l'outil Bobine au-dessus de son nœud.

Un segment de fil de liaison correspond à un seul élément horizontal ou vertical d'un fil de liaison. Un coude sur un fil de liaison correspond à la jonction de deux segments. L'emplacement sur lequel trois ou quatre segments de fil de liaison se rejoignent correspond à une jonction. Une branche d'un fil de liaison contient tous les segments de fil de liaison de jonction vers jonction, de terminal vers jonction ou de terminal vers terminal si aucune jonction n'existe entre eux. La figure 5-1 présente un segment de fil de liaison, un coude et une jonction.

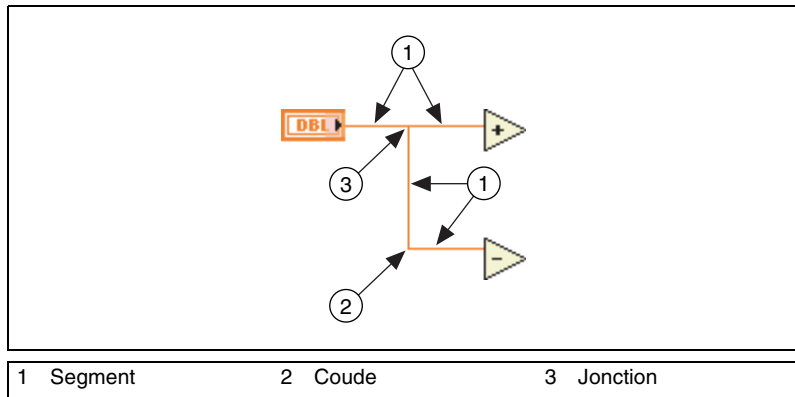


Figure 5-1. Segment de fil de liaison, coude et jonction

Vous pouvez câbler des terminaux verticalement ou horizontalement en fonction de la direction dans laquelle vous déplacez d'abord l'outil Bobine. Le fil de liaison se connecte au milieu du terminal, quel que soit l'emplacement du terminal sur lequel vous cliquez. Après avoir cliqué sur le terminal, basculez entre les directions horizontales et verticales en appuyant sur la barre espace.

Pendant que vous câblez un terminal, coudez une fois le fil de liaison à 90 degrés en déplaçant le curseur soit verticalement, soit horizontalement. Pour couder un fil de liaison dans plusieurs directions, cliquez sur le bouton de la souris afin de poser le fil et faites-le glisser dans la nouvelle direction. Vous pouvez à plusieurs reprises poser le fil et le faire glisser dans de nouvelles directions.

Pour annuler le dernier point où vous avez posé le fil de liaison, appuyez sur la touche <Maj> et cliquez n'importe où sur le diagramme.

(Macintosh) Appuyez sur la touche <Option> et cliquez.
(UNIX et Linux) Cliquez avec le bouton central de la souris.

Quand des fils de liaison se croisent, un petit espace apparaît sur le premier fil tracé pour indiquer que ce dernier se trouve en dessous du deuxième fil.



Mise en garde Le croisement de fils de liaison peut encombrer le diagramme et rendre sa mise au point difficile.

Câblage automatique des objets

Si vous avez activé le câblage automatique, LabVIEW câble automatiquement les objets au fur et à mesure que vous les placez sur le diagramme. Vous pouvez aussi câbler automatiquement des objets déjà placés sur le diagramme. LabVIEW connecte les terminaux qui correspondent au mieux et ne prend pas en compte ceux qui n'offrent pas de correspondance.

Au fur et à mesure que vous déplacez un objet sélectionné plus près d'autres objets du diagramme, LabVIEW tire des fils de liaison temporaires pour visualiser vos connexions valides. Quand vous relâchez le bouton de la souris pour positionner l'objet sur le diagramme, LabVIEW connecte automatiquement les fils de liaison.

Basculez en câblage automatique en appuyant sur la barre d'espace pendant que vous déplacez un objet en utilisant l'outil Flèche.

Par défaut, le câblage automatique est activé quand vous sélectionnez un objet à partir de la palette de **Fonctions** ou quand vous en copiez un qui se trouve déjà sur le diagramme en appuyant sur la touche <Ctrl> et en le faisant glisser. Le câblage automatique est désactivé par défaut quand vous utilisez l'outil Flèche pour déplacer un objet déjà présent sur le diagramme.

(Macintosh) Appuyez sur la touche <Option>. **(Sun)** Appuyez sur la touche <Meta>. **(Linux)** Appuyez sur la touche <Alt>.

Câblage manuel des objets

Utilisez l'outil Bobine pour connecter manuellement les terminaux sur un nœud du diagramme aux terminaux d'un autre nœud du diagramme. Le point du curseur de l'outil correspond à l'extrémité de la bobine de fil. Quand vous déplacez l'outil Bobine au-dessus d'un nœud, le terminal clignote et le nom du terminal apparaît dans une info-bulle, sur les fonctions et les VIs.

Sélectionnez **Aide»Aide contextuelle** pour afficher la fenêtre d'**Aide contextuelle** qui répertorie chaque terminal du VI ou de la fonction pour vous présenter exactement l'emplacement sur lequel les fils de liaison doivent être connectés. La fenêtre d'**Aide contextuelle** n'affiche pas les terminaux pour les fonctions extensibles, comme la fonction Construire un tableau.

Sélection des fils de liaison

Sélectionnez des fils de liaison à l'aide de l'outil Flèche en cliquant une fois, deux fois ou trois fois dessus. Cliquer une seule fois sur un fil de liaison sélectionne un seul segment de ce fil. Cliquer deux fois sur un fil de liaison sélectionne une branche. Cliquer trois fois sur un fil de liaison sélectionne le fil de liaison tout entier.

Suppression des fils de liaison brisés

Un fil de liaison brisé apparaît comme une ligne en pointillé. Les fils de liaison brisés apparaissent pour de multiples raisons, notamment lorsque vous essayez de câbler deux objets dont les types de données sont incompatibles. Déplacez l'outil Bobine au-dessus d'un fil de liaison brisé pour visualiser une info-bulle expliquant pourquoi le fil de liaison est brisé. Cliquez trois fois sur le fil de liaison avec l'outil Flèche et appuyez sur la touche <Suppr> pour supprimer un fil de liaison brisé.



Remarque Ne confondez pas le fil en tiret noir avec le fil pointillé vert. Un fil pointillé vert représente un type de données booléen.

Vous pouvez supprimer tous les fils de liaison en sélectionnant **Édition»Supprimer les fils brisés**.



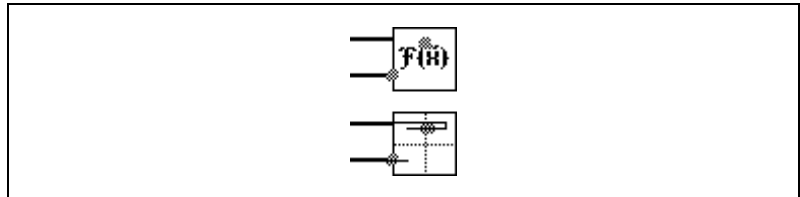
Mise en garde Soyez prudent quand vous supprimez tous les fils de liaison brisés. Quelquefois un fil de liaison apparaît brisé parce que vous n'avez pas terminé le câblage du diagramme.

Points de coercition

Les points de coercition apparaissent sur les nœuds d'un diagramme pour vous indiquer que vous avez câblé ensemble deux types de données différents. Le point signifie que LabVIEW a converti les valeurs transmises dans le nœud en une représentation différente. Par exemple, si vous câblez une commande avec la valeur 3,02 représentant un flottant à double précision sur un indicateur représentant un entier, un point de coercition apparaît sur l'indicateur et ce dernier devient 3.

Le diagramme positionne un point de coercition sur le bord d'un terminal dans lequel se produit la conversion pour indiquer que la conversion numérique automatique s'est effectuée.

Les VIs et les fonctions étant susceptibles de posséder de nombreux terminaux, un point de coercition peut apparaître à l'intérieur d'une icône si vous effectuez le câblage d'un terminal à un autre, comme l'illustre l'exemple suivant.



Les points de coercition peuvent être à l'origine de l'accroissement de l'utilisation de la mémoire d'un VI et augmenter sa durée d'exécution. Essayez de conserver la cohérence des types de données dans vos VIs.

VIs et fonctions polymorphes

Les VIs et les fonctions polymorphes peuvent s'adapter pour recevoir en entrée des données de types différents. La majorité des structures LabVIEW sont polymorphes, comme le sont certains VIs et fonctions.

VIs polymorphes

Les VIs polymorphes acceptent différents types de données pour un terminal de sortie ou d'entrée unique. Un VI polymorphe correspond à un ensemble de sous-VIs avec les mêmes modèles de connecteur. Chaque sous-VI correspond à une instance du VI polymorphe.

Par exemple, le VI Lire une clé est polymorphe. Son terminal **valeur par défaut** accepte des données booléennes, numériques à virgule flottante double précision, numériques entiers signés 32 bits, de chemin, de chaîne ou des données numériques entiers 32 bits non signés.

Les types de données que vous connectez aux terminaux d'entrée d'un VI polymorphe détermine l'instance à utiliser. Si un VI polymorphe ne contient pas un sous-VI pour ce type de données, un fil de liaison brisé apparaît. Vous pouvez sélectionner l'instance par défaut en cliquant avec le bouton droit de la souris sur le VI polymorphe et en sélectionnant **Sélectionner le type** dans le menu local, puis le sous-VI.

Construction de VIs polymorphes

Construisez vos propres VIs polymorphes quand vous réalisez la même opération sur des types différents de données. Vous ne pouvez construire des VIs polymorphes que dans la version Système de développement professionnel de LabVIEW.

Par exemple, si vous voulez réaliser la même opération mathématique sur une valeur numérique à virgule flottante simple précision, sur un tableau de valeurs numériques ou sur une waveform, vous pouvez créer trois VIs séparés : Calcul nombre, Calcul tableau et Calcul waveform. Quand vous avez besoin de réaliser l'opération, vous placez l'un de ces trois VIs dans le diagramme, en fonction du type de données que vous utilisez en entrée.

Au lieu de placer manuellement une version du VI sur le diagramme, vous pouvez créer et utiliser un VI polymorphe unique. Le VI polymorphe Calcul contient trois instances du VI, comme l'illustre la figure 5-2.

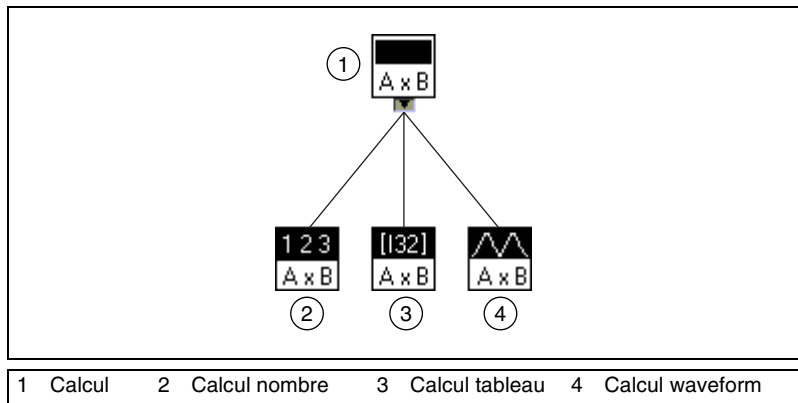


Figure 5-2. Exemple d'un VI polymorphe

Le VI Calcul appelle l'instance correcte du VI en se basant sur le type de données que vous câblez sur le sous-VI du diagramme, comme l'illustre la figure 5-3.

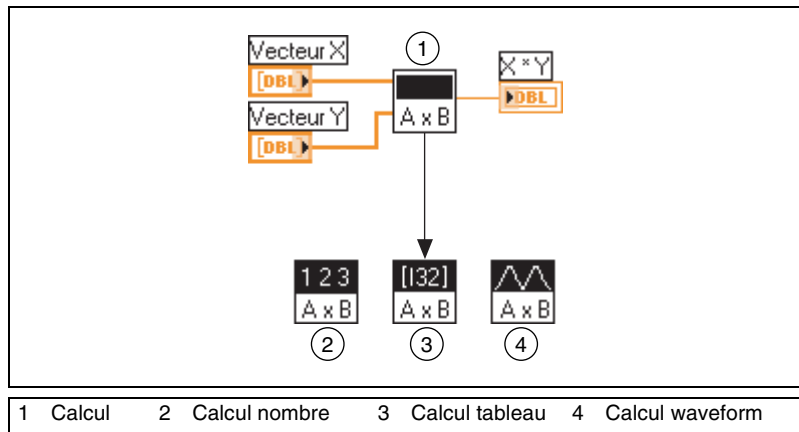


Figure 5-3. VI polymorphe appelant un sous-VI

Les VIs polymorphes diffèrent de la majorité des VIs dans la mesure où ils ne possèdent ni face-avant, ni diagramme.

Veillez prendre en considération les problèmes suivants quand vous construisez des VIs polymorphes :

- Tous les VIs que vous incluez dans le VI polymorphe doivent avoir le même modèle de connecteur parce que le connecteur du VI polymorphe correspond à celui des VIs que vous utilisez pour créer le VI polymorphe.
- Les entrées et les sorties sur le connecteur de chaque instance de VI doivent correspondre aux entrées et aux sorties du connecteur du VI polymorphe.
- Les VIs que vous utilisez pour construire des VIs polymorphes ne doivent pas se composer des mêmes fonctions et sous-VIs.
- Les faces-avant des VIs n'ont pas toutes besoin de posséder le même nombre d'objets. Néanmoins, chaque face-avant doit avoir au moins le même nombre de commandes et d'indicateurs qui constituent le connecteur du VI polymorphe.
- Vous pouvez créer une icône pour un VI polymorphe.
- Vous ne pouvez pas utiliser des VIs polymorphes dans d'autres VIs polymorphes.

Quand vous créez une documentation complète pour un VI qui inclut un sous-VI polymorphe, le VI polymorphe et les VIs qu'il invoque apparaissent dans la **Liste des sous-VIs**.

Fonctions polymorphes

Les fonctions sont polymorphes à des degrés variables : aucune, certaines ou toutes leurs sorties peuvent être polymorphes. Certaines entrées de fonctions acceptent des valeurs numériques ou booléennes. Certaines acceptent des valeurs numériques ou des chaînes. Certaines acceptent non seulement des valeurs numériques scalaires mais aussi des tableaux de valeurs numériques, des clusters de valeurs numériques et ainsi de suite. Certaines acceptent uniquement des tableaux à une seule dimension, même si les éléments de tableau peuvent être de n'importe quel type. Certaines fonctions acceptent tous les types de données y compris les nombres complexes. Reportez-vous à l'annexe B, *Fonctions polymorphes*, pour obtenir de plus amples informations à propos des fonctions polymorphes.

Manipulation de données variant

Les données variant ne sont pas conformes à des données d'un type spécifique et contiennent des attributs, tels que des noms et des unités de voies. LabVIEW représente les données variant avec le type de données variant. Le type de données variant diffère des autres types de données, car il inclut le nom de la commande ou de l'indicateur, des informations sur le type de données et les données elles-mêmes.

Utilisez les fonctions variant situées sur la palette **Fonctions»Avancé» Manipulation de données»Variant** pour créer et manipuler des données de type variant. Vous pouvez convertir tout type de données LabVIEW en un type de données variant afin d'utiliser ces dernières avec d'autres VIs et fonctions. Par exemple, si vous convertissez une chaîne en données variant, le type de données variant stocke le texte en indiquant que le texte est une chaîne.

Utilisez le type de données variant quand il s'avère important de manipuler des données indépendamment de leur type. Vous pouvez aussi représenter les données indépendamment du type sans utiliser le type de données variant en aplatissant les données dans les chaînes. Par exemple, la méthode **Valeurs des commandes** retourne les informations à propos des commandes et des indicateurs provenant d'un VI sous la forme d'un tableau de clusters. Chaque cluster dans le tableau contient le type des données et la valeur de chaque commande ou indicateur. Le cluster contient le nom de la commande ou de l'indicateur sous la forme d'une chaîne, le type de

données sous la forme d'un tableau d'entiers de 16 bits et les données sous la forme de chaînes aplaties, comme l'illustre la figure 5-4.

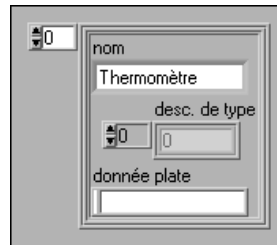


Figure 5-4. Cluster de données numériques aplaties

Néanmoins, l'utilisation de données aplaties impose certaines limites, parce que LabVIEW ne peut pas contraindre des données aplaties. De plus, essayer de redresser un entier aplati sous la forme d'un nombre avec un flottant de précision étendue se traduit par un échec. Utilisez le type de données variant pour travailler des données indépendamment du type de données au lieu d'aplatir les données. Reportez-vous à la section *Flattened Data* de la note d'application *LabVIEW Data Storage* pour obtenir de plus amples informations quant à l'aplatissement et au redressement des données.

Vous pouvez ajouter des attributs afin d'identifier ultérieurement les données variant. Par exemple, un attribut du type de données variant peut identifier de quelle voie de périphérique d'acquisition de données celles-ci sont issues.

Les données variant sont également utiles quand vous écrivez ou lisez à partir de la mémoire dans LabVIEW et quand vous effectuez des opérations sur des piles (dernier entré, premier sorti), files d'attente (premier entré, premier sorti), des buffers intelligents ou des arbres. Ces sortes d'opérations traitent les données indépendamment de leur type.

Unités numériques et vérification de type stricte

Vous pouvez associer des unités physiques de mesures, comme des mètres ou des kilomètres par seconde, à n'importe quel contrôle numérique ayant une représentation de type flottant.

Les unités d'une commande apparaissent sur une étiquette séparée liée à la commande, appelée étiquette d'unité. Affichez l'étiquette d'unité en cliquant avec le bouton droit de la souris sur la commande et en sélectionnant **Éléments visibles»Étiquette d'unité** à partir du menu local.

Quand LabVIEW affiche l'étiquette d'unité, vous pouvez entrer une unité en utilisant les abréviations standard telles que *m* pour mètres, *ft* pour pieds, *s* pour secondes et ainsi de suite.



Remarque Vous ne pouvez pas utiliser des unités dans les boîtes de calcul.

Unités et vérification de type stricte

Quand vous associez des unités à un objet, vous ne pouvez câbler que des objets ayant des unités compatibles. LabVIEW utilise une vérification de type stricte afin de s'assurer que les unités sont compatibles. Si vous câblez deux objets avec des unités incompatibles, LabVIEW retourne une erreur. Par exemple, LabVIEW retourne une erreur si vous câblez un objet avec le kilomètre comme type d'unité sur un objet dont le type d'unité est le litre, car le kilomètre est une unité de distance et le litre, une unité de volume.

La figure 5-5 présente le câblage d'objets avec des unités compatibles. Dans cette figure, LabVIEW met automatiquement à l'échelle l'indicateur de **distance** afin d'afficher des kilomètres et non des mètres, car le kilomètre est l'unité de l'indicateur.

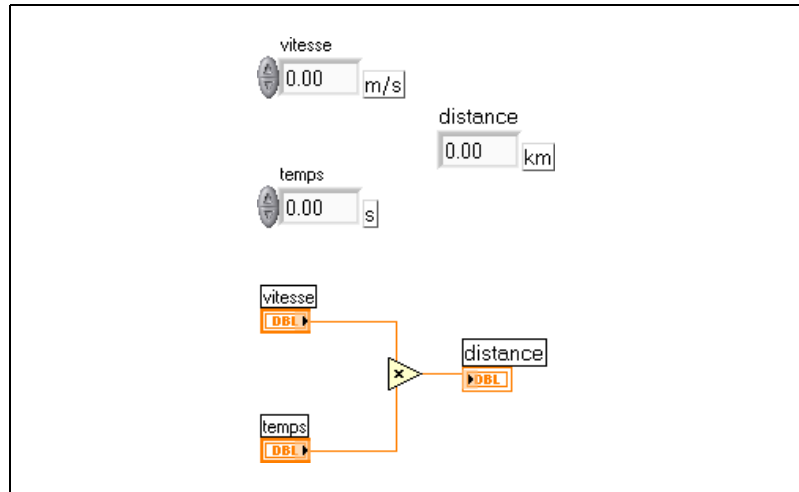


Figure 5-5. Câblage d'objets avec des unités compatibles

Une erreur se produit sur la figure 5-6 car les unités de **distance** sont exprimées en secondes. Pour corriger l'erreur, modifiez les secondes en une unité de distance, le kilomètre par exemple, comme l'illustre la figure 5-5.

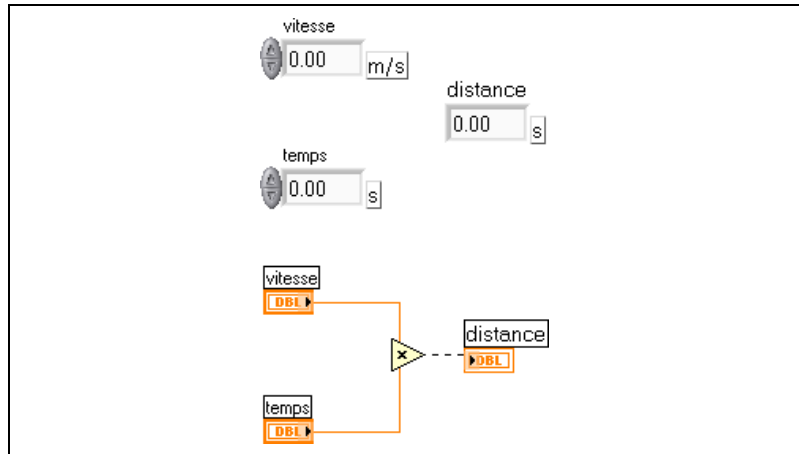


Figure 5-6. La connexion de sources avec des unités incompatibles se traduit par des fils de liaison brisés

Pour voir l'erreur, amenez l'outil Bobine sur le fil de liaison brisé jusqu'à ce qu'un conseil apparaisse ou cliquez avec le bouton droit sur le fil de liaison brisé et sélectionnez **Liste des erreurs** dans le menu local. La fenêtre **Liste des erreurs** répertorie l'erreur suivante :

Vous avez connecté des types de données numériques présentant des unités incompatibles.

Certains VIs et fonctions sont ambigus en ce qui concerne les unités. Vous ne pouvez pas utiliser ces VIs et fonctions avec d'autres terminaux qui ont des unités. Par exemple, la fonction d'incrémentement est ambiguë en ce qui concerne les unités. Si vous utilisez des unités de distance, la fonction d'incrémentement est incapable de dire s'il faut ajouter un mètre, un kilomètre ou un pied. C'est en raison de cette ambiguïté que vous ne pouvez pas utiliser la fonction d'incrémentement, ni d'autres fonctions qui incrémentent ou décrémentent des valeurs, avec des données ayant des unités associées.

Pour éviter toute ambiguïté dans cet exemple, utilisez une constante numérique avec l'unité adéquate ainsi que la fonction Ajouter pour créer votre propre fonction d'incrémement d'unités, comme l'illustre la figure 5-7.

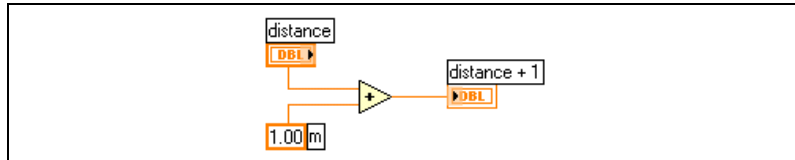


Figure 5-7. Création d'une fonction d'incrémement avec des unités

Flux des données dans le diagramme

LabVIEW suit un modèle de flux de données pour l'exécution des VIs. Un nœud du diagramme s'exécute lorsque toutes ses entrées sont disponibles. Lorsqu'un nœud termine son exécution, il fournit des données à ses terminaux de sortie et transmet ces données au nœud suivant dans le chemin de flux de données.

Visual Basic, C++, Java et la plupart des autres langages de programmation textuels suivent un modèle d'exécution séquentielle. Dans la programmation séquentielle, l'ordre séquentiel des éléments du programme détermine l'ordre d'exécution du programme.

Comme LabVIEW utilise un modèle à flux de données plutôt qu'un modèle séquentiel, vous pouvez créer des diagrammes ayant des opérations simultanées. LabVIEW est un système multitâche et multithread capable d'exécuter plusieurs threads et plusieurs VIs simultanément.

Reportez-vous à la note d'application *Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability*, pour obtenir de plus amples informations sur le traitement simultané de plusieurs tâches dans LabVIEW.

Dépendance des données et dépendance des données artificielle

Dans le modèle de programmation séquentielle, l'exécution dépend des instructions. En revanche, dans le modèle de programmation à flux de données, ce sont les données qui déterminent l'ordre d'exécution. Un nœud qui reçoit des données d'un autre nœud ne s'exécute que lorsque le premier nœud a terminé son exécution.

Des nœuds de diagramme qui ne sont pas connectés par des fils de liaison peuvent s'exécuter dans n'importe quel ordre. Bien que le manuel *LabVIEW Development Guidelines* vous recommande d'utiliser une présentation de gauche à droite et de haut en bas, les nœuds ne s'exécutent pas nécessairement dans cet ordre.

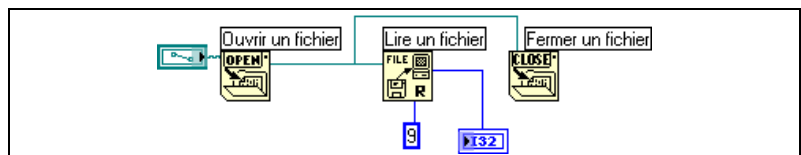
Utilisez la structure Séquence pour contrôler l'ordre d'exécution lorsqu'il n'existe pas de dépendance de données naturelle. Reportez-vous à la section *Structures Séquence* du chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour de plus amples informations sur les structures Séquence. Vous pouvez également utiliser des paramètres dupliqués pour contrôler l'ordre d'exécution. Reportez-vous à la section *Paramètres dupliqués* du chapitre 13, *E/S sur fichiers*, pour de plus amples informations sur les paramètres dupliqués.

Une autre façon de contrôler l'ordre d'exécution est de créer une dépendance de données artificielle, dans laquelle le nœud récepteur n'utilise pas vraiment les données reçues sur ses terminaux d'entrée. À la place, le nœud récepteur utilise l'arrivée de données pour déclencher son exécution. Reportez-vous au VI Timing Template (data dep) dans `examples\general\structs.llb` pour étudier un exemple de dépendance de données artificielle.

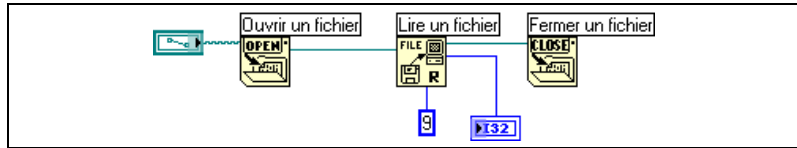
Dépendance de données manquante

Ne présumez pas que l'exécution se déroule de gauche à droite et de haut en bas lorsqu'il n'y a pas de dépendance de données. Assurez-vous d'avoir clairement défini la séquence d'événements lorsque cela est nécessaire en câblant le flux des données.

Dans l'exemple suivant, il n'existe pas de dépendance de données entre la fonction Lire un fichier et la fonction Fermer un fichier parce que la fonction Lire un fichier n'est pas câblée à la fonction Fermer un fichier. Cet exemple peut générer des résultats inattendus, il est en effet impossible de savoir quel VI s'exécute en premier. Si la fonction Fermer un fichier est la première à s'exécuter, la fonction Lire un fichier ne fonctionne pas.



Le diagramme suivant établit une dépendance en câblant une sortie de la fonction Lire un fichier à la fonction Fermer un fichier. La fonction Fermer un fichier ne s'exécute pas tant qu'elle n'a pas reçu la sortie de la fonction Lire un fichier.



Flux des données et gestion de mémoire

L'exécution par flux de données facilite la gestion de mémoire par rapport au modèle d'exécution séquentielle. Dans LabVIEW, vous n'allouez pas de variables et vous ne leur affectez pas de valeur. À la place, vous créez un diagramme avec des fils de liaison qui représentent la transition des données.

Les VIs et les fonctions qui génèrent des données allouent automatiquement de la mémoire à ces données. Lorsque le VI ou la fonction n'utilise plus les données, LabVIEW désalloue la mémoire associée. Lorsque vous ajoutez de nouvelles données à un tableau ou à une chaîne, LabVIEW alloue assez de mémoire pour gérer les nouvelles données.

Comme LabVIEW gère la mémoire automatiquement, vous ne contrôlez pas les phases d'allocation et de désallocation. Si votre VI utilise de grandes quantités de données, vous devez comprendre le principe d'allocation de mémoire. Comprendre ces principes peut en effet vous aider à écrire des VIs avec des besoins en mémoire largement inférieurs. Minimiser l'utilisation de la mémoire peut vous aider à augmenter la vitesse d'exécution des VIs. Reportez-vous à la note d'application *LabVIEW Performance and Memory Management*, pour obtenir de plus amples informations sur l'allocation mémoire.

Conception du diagramme

Utilisez les conseils suivants pour concevoir vos diagrammes :

- Utilisez une disposition de gauche à droite et de haut en bas. Bien que les positions des éléments du diagramme ne déterminent pas l'ordre d'exécution, évitez de câbler de droite à gauche pour garder le diagramme organisé et facile à lire. Seuls les fils de liaison et les structures déterminent l'ordre d'exécution.
- Évitez de créer un diagramme qui occupe plus d'un ou de deux écrans. Si un diagramme devient large et complexe, il peut être difficile à comprendre ou à mettre au point.
- Voyez si vous pouvez réutiliser certaines composantes du diagramme dans d'autres VIs ou si une section du diagramme peut être utilisée comme une composante logique. Si c'est le cas, divisez le diagramme en sous-VIs qui remplissent des tâches spécifiques. Utiliser des sous-VIs vous aide à gérer des changements et à mettre rapidement au point les diagrammes. Reportez-vous à la section *Sous-VIs* du chapitre 7, *Création de VIs et de sous-VIs*, pour de plus amples informations sur les sous-VIs.
- Utilisez les VIs, les fonctions et les paramètres de gestion d'erreur pour gérer les erreurs dans le diagramme. Reportez-vous à la section *Recherche et gestion d'erreur* du chapitre 6, *Exécution et mise au point des VIs*, pour obtenir de plus amples informations sur la gestion des erreurs.
- Améliorez la présentation du diagramme en câblant de manière efficace. Une mauvaise organisation des fils de liaison ne produit pas forcément d'erreur, mais rend le diagramme difficile à lire ou à mettre au point ; elle peut aussi parfois laisser des doutes sur ce que fait réellement le VI.
- Évitez de câbler sous un cadre de structure ou entre des objets qui se chevauchent, parce que LabVIEW peut masquer certains segments du fil de liaison résultant.
- Évitez de mettre des objets par-dessus des fils de liaison. Les fils de liaison ne connectent que les objets sur lesquels vous avez cliqué pour fixer le fil. Mettre un terminal ou une icône par-dessus un fil de liaison fait croire qu'une connexion existe alors que ce n'est pas le cas.

- Utilisez l'outil Pinceau et cliquez avec le bouton droit sur l'espace de travail du diagramme pour ajouter ou changer la couleur de l'espace de travail.
- Augmentez l'espace entre des objets entassés ou groupés trop près en appuyant sur la touche <Ctrl> et en utilisant l'outil Flèche pour cliquer sur l'espace de travail du diagramme. Tout en maintenant ces touches enfoncées, faites glisser l'outil jusqu'à l'obtention d'un rectangle de la taille désirée.

(Macintosh) Appuyez sur la touche <Option>. **(Sun)** Appuyez sur la touche <Meta>. **(Linux)** Appuyez sur la touche <Alt>.

Un rectangle entouré de pointillés définit l'endroit où cet espace va être inséré. Relâchez la touche.

Reportez-vous au manuel *LabVIEW Development Guidelines* pour obtenir plus d'informations sur le diagramme.

Exécution et mise au point des VIs

Pour exécuter un VI, vous devez câbler tous les terminaux requis des sous-VIs, fonctions et structures aux types de données adéquats. Quelquefois, un VI produit des données ou s'exécute de manière inattendue. Vous pouvez utiliser LabVIEW pour configurer l'exécution d'un VI et pour identifier des problèmes liés à l'organisation du diagramme ou à la transmission des données dans le diagramme.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir plus d'informations sur la mise au point des VIs.

Exécution des VIs



Exécuter un VI exécute l'opération pour laquelle vous avez conçu le VI. Vous pouvez exécuter un VI si le bouton **Exécuter** sur la barre d'outils apparaît comme une flèche blanche pleine, illustrée à gauche. La flèche blanche indique également que vous pouvez utiliser le VI comme un sous-VI si vous créez un connecteur pour le VI.

Un VI s'exécute lorsque vous cliquez sur les boutons **Exécuter** ou **Exécuter en continu**, ou sur les boutons de mode pas à pas de la barre d'outils du diagramme. Cliquer sur le bouton **Exécuter** exécute le VI une fois. Le VI s'arrête lorsque le flux de données s'achève. Cliquer sur le bouton **Exécuter en continu** exécute en continu le VI jusqu'à ce que vous cliquiez sur le bouton **Stop** du VI. Cliquer sur les boutons de mode pas à pas exécute le VI par incréments. Reportez-vous à la section *Mode pas à pas* de ce chapitre pour plus d'informations sur l'utilisation des boutons d'exécution en mode pas à pas pour mettre au point un VI.



Remarque Évitez d'utiliser le bouton **Abandonner l'exécution** pour stopper un VI. Laissez le VI achever son flux de données ou concevez une méthode pour arrêter le VI par programmation. Ainsi, le VI est dans un état connu. Par exemple, placez un bouton sur la face-avant qui arrête le VI quand vous cliquez dessus.

Configuration de l'exécution d'un VI

Sélectionnez **Fichier»Propriétés du VI...** et sélectionnez **Exécution** dans le menu déroulant **Catégorie** pour configurer l'exécution d'un VI. Ainsi, vous pouvez configurer un VI pour qu'il s'exécute à l'ouverture ou qu'il se mette en mode pause lorsqu'il est appelé en tant que sous-VI. Vous pouvez aussi configurer le VI pour qu'il s'exécute selon diverses priorités. Par exemple, s'il est capital qu'un VI s'exécute sans attendre la fin d'une autre opération, configurez son exécution sur priorité critique (la plus haute). Reportez-vous à la note d'application *Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability* pour obtenir plus de détails sur la création de VIs multithread. Reportez-vous au chapitre 15, *Personnalisation des VIs*, pour obtenir de plus amples informations sur la configuration de l'exécution des VIs.

Correction des VIs brisés



Si un VI ne s'exécute pas, il s'agit d'un VI brisé ou non exécutable. Le bouton **Exécuter** apparaît souvent brisé, comme indiqué sur la gauche, lorsque vous créez ou éditez un VI. S'il est toujours brisé lorsque vous terminez de câbler le diagramme, le VI est brisé et il ne s'exécutera pas.

Recherche des causes pour les VIs brisés

Cliquez sur le bouton **Exécuter** ou sélectionnez **Fenêtre»Liste des erreurs** pour déterminer pourquoi un VI est brisé. La fenêtre **Liste des erreurs** liste toutes les erreurs. La section **Liste des VIs** liste les noms de tous les VIs en mémoire qui contiennent des erreurs. La section **erreurs et mises en garde** liste les erreurs et mises en garde du VI sélectionné dans la section **Liste des VIs**. La section **Détails** décrit les erreurs et, dans certains cas, indique comment les corriger ou trouver davantage d'informations pour les résoudre. Sélectionnez **Aide** pour ouvrir un fichier d'aide en ligne listant les erreurs LabVIEW et leurs descriptions.

Cliquez sur le bouton **Visualiser l'erreur** ou double-cliquez sur la description de l'erreur pour afficher le diagramme ou la face-avant et mettre en surbrillance l'objet contenant l'erreur.



La barre d'outils inclut le bouton **Mise en garde** sur la gauche, si un VI comporte une mise en garde et que vous avez coché **Visualiser les mises en garde** dans la fenêtre **Liste des erreurs**.

Vous pouvez configurer LabVIEW pour toujours afficher les mises en garde dans la fenêtre **Liste des erreurs** en sélectionnant **Outils» Options...**, puis **Mise au point** dans le menu déroulant du haut, et en cochant **Visualiser les mises en garde dans la liste des erreurs**. Vous pouvez également effectuer ce changement dans la fenêtre **Liste des erreurs** ouverte et voir immédiatement le changement.

Les mises en garde ne vous empêchent pas d'exécuter un VI. Elles sont simplement destinées à vous indiquer d'éventuels problèmes dans les VIs.

Causes courantes de VIs brisés

Lorsqu'un VI se brise pendant que vous l'éditez, les causes sont en général les suivantes :

- Le diagramme contient un fil de liaison brisé parce que vous avez câblé des données incompatibles ou un segment est déconnecté. Reportez-vous à la section *Utilisation des fils de liaison pour relier les objets du diagramme* du chapitre 5, *Construction du diagramme*, pour plus d'informations sur le câblage des objets dans le diagramme.
- Il manque un câblage nécessaire à un terminal du diagramme. Sélectionnez **Aide»Aide contextuelle** ou reportez-vous à l'*Aide LabVIEW* pour déterminer quels sont les paramètres requis par un nœud du diagramme.
- Un sous-VI est brisé ou vous avez édité son connecteur après avoir placé son icône sur le diagramme du VI. Reportez-vous à la section *Sous-VIs* du chapitre 7, *Création de VIs et de sous-VIs*, pour de plus amples informations sur les sous-VIs.

Techniques de mise au point

Si un VI n'est pas brisé, mais que vous recevez des données inattendues, vous pouvez utiliser les techniques suivantes pour identifier et corriger les problèmes relatifs au VI ou au flux des données dans le diagramme :

- Câblez les paramètres d'**entrée d'erreur** et de **sortie d'erreur** au bas de la plupart des VIs intégrés et des fonctions. Ces paramètres détectent les erreurs rencontrées sur chaque nœud du diagramme et indiquent si une erreur s'est produite et, le cas échéant, à quel endroit. Vous pouvez également utiliser ces paramètres dans les VIs que vous construisez. Reportez-vous à la section [Gestion d'erreur](#) de ce chapitre pour plus d'informations sur l'utilisation de ces paramètres.
- Utilisez le mode Animation pour observer le déplacement des données dans le diagramme.
- Exécutez le VI pas à pas pour observer chaque action du VI sur le diagramme.
- Utilisez l'outil Sonde pour vérifier les valeurs intermédiaires lors de l'exécution.
- Utilisez des points d'arrêt pour mettre l'exécution sur pause et entrer en mode pas à pas, ou insérer des sondes.
- Arrêtez momentanément l'exécution d'un sous-VI pour éditer des valeurs de commandes et indicateurs, pour contrôler le nombre d'exécutions, ou pour revenir au début de l'exécution du sous-VI.
- Désactivez à l'aide d'une structure Condition une section particulière du diagramme pour déterminer si le VI fonctionne mieux sans elle.

Mode Animation



Visualisez l'animation de l'exécution du diagramme en cliquant sur le bouton **Animer l'exécution**, présenté sur la gauche. Le mode Animation présente le mouvement des données sur le diagramme d'un nœud à un autre en utilisant des "bulles" qui se déplacent le long des fils de liaison. Utilisez le mode Animation avec le mode pas à pas pour observer comment les données se déplacent de nœud à nœud dans un VI.



Remarque Le mode Animation réduit énormément la vitesse d'exécution du VI.

Si le cluster **sortie d'erreur** rapporte une erreur, la valeur de l'erreur apparaît à côté de **sortie d'erreur** avec un cadre rouge. Si aucune erreur ne survient, **OK** apparaît à côté de **sortie d'erreur** avec un cadre vert.

Reportez-vous à la section *Clusters d'erreur* de ce chapitre, pour plus d'informations sur les clusters d'erreur.

Mode pas à pas



Exécutez le VI pas à pas pour observer chaque action du VI sur le diagramme pendant l'exécution. Les boutons du mode pas à pas n'affectent que l'exécution d'un VI ou d'un sous-VI en mode pas à pas. Accédez au mode pas à pas en cliquant sur le bouton **Exécuter sans détailler** ou **Exécuter de façon détaillée** sur la barre d'outils du diagramme. Déplacez le curseur sur le bouton **Exécuter sans détailler**, **Exécuter de façon détaillée** ou **Terminer l'exécution** pour afficher une info-bulle qui décrit l'étape suivante si vous cliquez sur ce bouton. Vous pouvez utiliser le mode pas à pas dans des sous-VIs ou les exécuter normalement.



Si vous utilisez le mode pas à pas dans un VI avec le mode Animation activé, un symbole d'exécution, présenté sur la gauche, apparaît sur les icônes des sous-VIs en cours d'exécution.

Outil Sonde



Utilisez l'outil Sonde, présenté sur la gauche, pour vérifier des valeurs intermédiaires sur un fil de liaison, lors de l'exécution d'un VI. Utilisez l'outil Sonde si votre diagramme est compliqué, avec une série d'opérations, n'importe laquelle d'entre elles pouvant retourner des données incorrectes. Utilisez l'outil Sonde avec le mode Animation, le mode pas à pas et des points d'arrêt pour déterminer si des données sont incorrectes et, le cas échéant, où se situe le problème. Si des données sont disponibles, la sonde se met immédiatement à jour pendant le mode pas à pas ou lorsque vous effectuez une pause à un point d'arrêt. Lorsque l'exécution effectue une pause à un nœud, à cause du mode pas à pas ou d'un point d'arrêt, vous pouvez également sonder le fil de liaison qui vient de s'exécuter pour voir la valeur qui a traversé ce fil.

Vous pouvez également créer une sonde personnalisée pour spécifier quel indicateur utiliser pour afficher les données sondées. Par exemple, si vous observez des données numériques, vous pouvez, si vous le désirez, voir ces données dans un graphe déroulant à l'intérieur de la sonde. Effectuez un clic droit sur un fil de liaison et sélectionnez **Sonde personnalisée** ; spécifiez ensuite l'indicateur que vous voulez utiliser.

Points d'arrêt



Utilisez l'outil Point d'arrêt, présenté sur la gauche, pour placer un point d'arrêt sur un VI, un nœud ou un fil de liaison sur le diagramme et interrompre momentanément l'exécution à cet emplacement. Lorsque vous définissez un point d'arrêt sur un fil de liaison, l'exécution effectue une pause après le passage des données dans le fil de liaison. Insérez un point d'arrêt sur l'espace de travail du diagramme pour mettre l'exécution sur pause après l'exécution de tous les nœuds du diagramme.

Quand un VI effectue une pause à un point d'arrêt, LabVIEW amène le diagramme vers l'avant et fait clignoter le nœud ou le fil de liaison qui contient le point d'arrêt. Quand vous déplacez le curseur au-dessus d'un point d'arrêt existant, la zone noire du curseur correspondant à l'outil Point d'arrêt devient blanche.

Lorsque vous atteignez un point d'arrêt pendant l'exécution, vous pouvez choisir une des actions suivantes :

- Exécuter en mode pas à pas avec les boutons du mode pas à pas.
- Sonder les fils de liaison pour vérifier les valeurs intermédiaires.
- Continuer l'exécution jusqu'au point d'arrêt suivant ou jusqu'à ce que le VI termine l'exécution.

LabVIEW enregistre les points d'arrêt avec un VI, mais ceux-ci ne sont activés que lors de l'exécution.

Suspension de l'exécution

Arrêtez momentanément l'exécution d'un sous-VI pour éditer des valeurs de commandes et indicateurs, pour contrôler le nombre d'exécutions avant le retour à l'appelant ou pour revenir au début de l'exécution du sous-VI. Si vous le décidez, tous les appels à un sous-VI peuvent démarrer avec l'exécution suspendue ou vous pouvez interrompre un appel spécifique à un sous-VI.

Pour suspendre tous les appels à un sous-VI, ouvrez le sous-VI et sélectionnez **Exécution»Interrompre l'exécution à l'appel**. Le sous-VI s'interrompt automatiquement lorsqu'un autre VI l'appelle. Si vous sélectionnez cet élément de menu avec le mode pas à pas, le sous-VI ne s'interrompt pas immédiatement. Le sous-VI s'interrompt lorsqu'il est appelé.

Pour suspendre un appel de sous-VI spécifique, cliquez avec le bouton droit sur le nœud de sous-VI sur le diagramme et sélectionnez **Configuration du nœud du sous-VI** dans le menu local. Cochez **Interrompre l'exécution à l'appel** pour suspendre l'exécution uniquement à cette instance du sous-VI.

La **fenêtre de hiérarchie**, que vous affichez en sélectionnant **Parcourir» Hiérarchie du VI**, indique si un VI est en pause ou en mode interruption. Un symbole de flèche indique un VI qui s'exécute de manière normale ou en mode pas à pas. Un symbole de pause indique un VI en pause ou en mode interruption. Un symbole de pause vert, ou un symbole vide noir et blanc, indique un VI qui s'est mis en pause lorsqu'il a été appelé. Un symbole de pause rouge, ou un symbole noir et blanc, indique un VI qui est actuellement sur pause. Un symbole en point d'exclamation indique que le sous-VI est suspendu. Un VI peut être à la fois suspendu et en pause.

Détermination de l'instance actuelle d'un sous-VI

Lorsque vous mettez un sous-VI en pause, le menu déroulant **Chaîne d'appel** liste la chaîne des appelants du VI de niveau principal au sous-VI. Il ne s'agit pas de la même liste qui s'affiche lorsque vous sélectionnez **Parcourir» VIs appelant ce VI**, qui liste tous les VIs appelants, qu'ils soient en cours d'exécution ou non. Utilisez le menu **Chaîne d'appel** pour déterminer l'instance actuelle du sous-VI si le diagramme contient plusieurs appels à ce sous-VI. Lorsque vous sélectionnez le VI appelant dans le menu **Chaîne d'appel**, son diagramme s'ouvre et LabVIEW met en surbrillance l'instance actuelle du sous-VI.

Mise “en commentaire” d'une section particulière du diagramme

Vous pouvez exécuter un VI avec une section du diagramme désactivée, de la même façon que vous pouvez “commenter” une section particulière du code dans un langage de programmation textuel. Il peut en effet être utile de désactiver une section du diagramme pour déterminer si le VI fonctionne mieux sans elle.

Pour cela, mettez la section que vous désirez désactiver à l'intérieur d'une structure Condition et utilisez une constante booléenne pour exécuter au choix l'une des deux conditions. Reportez-vous à la section [Structures Condition](#) du chapitre 8, [Boucles For et While et Structures Condition, Séquence et Événement](#), pour plus d'informations sur l'utilisation des structures Condition. Vous pouvez également créer une copie du VI et supprimer cette section du diagramme dans la copie. Détruisez ensuite la version du VI que vous décidez de ne pas utiliser.

Désactivation des outils de mise au point

Vous pouvez désactiver les outils de mise au point pour réduire les besoins en mémoire et augmenter légèrement les performances. Cliquez avec le bouton droit sur le connecteur et sélectionnez **Propriétés du VI**. Sélectionnez **Exécution** dans le menu déroulant **Catégorie** et désactivez **Autoriser la mise au point**.

Données indéfinies ou inattendues

Les données indéfinies, qui sont signalées par NaN (not a number) ou Inf (infini), invalident toutes les opérations ultérieures. Les opérations sur les flottants retournent les deux valeurs symboliques suivantes qui indiquent des calculs erronés ou des résultats impossibles :

- NaN (not a number) représente la valeur flottante générée par des opérations incorrectes, comme le calcul de la racine carrée d'un nombre négatif.
- Inf (infinity) représente la valeur flottante générée par des opérations illégales, comme diviser un nombre par zéro.

LabVIEW ne vérifie pas les conditions de dépassement des limites inférieures ou supérieures pour des valeurs d'entiers. Le dépassement des limites inférieures ou supérieures pour les nombres flottants est conforme à la norme IEEE 754, *Standard for Binary Floating-Point Arithmetic*.

Les opérations sur des nombres flottants propagent NaN et Inf fidèlement. Cependant, lorsque vous convertissez, implicitement ou explicitement, NaN ou Inf en entiers ou en valeurs booléennes, les valeurs ne signifient rien. Par exemple, diviser 1 par zéro produit Inf. Convertir Inf en un entier 16 bits produit la valeur 32767 , qui semble être une valeur normale. Reportez-vous à la section [Conversion numérique](#) de l'annexe B, *Fonctions polymorphes*, pour plus d'informations sur la conversion des valeurs numériques.

Avant de convertir des données en entiers, utilisez l'outil Sonde pour vérifier la validité des valeurs intermédiaires de type flottant. Vérifiez NaN en câblant la fonction de comparaison Pas un nombre/chemin/refnum ? à la valeur que vous pensez être incorrecte.

Données par défaut et valeurs inattendues dans des boucles

Les boucles For produisent des valeurs inattendues lorsque le compte d'itération est zéro pour une boucle For auto-indexée. Les boucles While produisent des données par défaut lorsque le registre à décalage n'est pas initialisé.

Boucles For

Les boucles For peuvent produire des valeurs inattendues si vous câblez 0 au terminal de décompte des itérations ou si vous câblez un tableau vide à l'entrée avec l'auto-indexation activée. La boucle ne s'exécute pas et les tunnels de sortie dont l'auto-indexation est désactivée contiennent des valeurs inattendues. Utilisez les registres à décalage pour transférer des valeurs à travers une boucle, que celle-ci s'exécute ou non.

Reportez-vous à la section *Structures des boucles For et While* du chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour plus d'informations sur les boucles For, l'auto-indexation et les registres à décalage.

Boucles While

Si vous n'initialisez pas un registre à décalage sur une boucle While, la sortie est la valeur par défaut du paramètre (0, FAUX, chaîne vide, etc.) ou la dernière valeur chargée dans le registre à décalage lors de la dernière exécution du VI.

Données par défaut dans les tableaux

L'indexation au-delà des limites d'un tableau produit la valeur par défaut pour le paramètre de l'élément du tableau. Vous pouvez utiliser la fonction Taille d'un tableau pour déterminer la taille du tableau. Reportez-vous à la section *Tableaux* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour plus d'informations sur les tableaux. Reportez-vous à la section *Auto-indexation des boucles* du chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour plus d'informations sur l'indexation. Vous pouvez, par inadvertance, indexer au-delà des limites d'un tableau en indexant un tableau après le dernier élément en utilisant une boucle While, en fournissant une valeur trop importante à l'entrée d'**indice** d'une fonction Indexer un tableau ou en fournissant un tableau vide à cette même fonction.

Éviter des données indéfinies

Ne vous fiez pas à des valeurs spéciales, comme NaN, Inf ou des tableaux vides, pour déterminer si un VI produit des données indéfinies. Au lieu de cela, confirmez que le VI produit des données définies en faisant générer une erreur par le VI s'il rencontre une situation qui produira probablement des données indéfinies.

Par exemple, si vous créez un VI qui utilise un tableau entrant pour auto-indexer une boucle For, déterminez l'action désirée du VI lorsque le tableau d'entrée est vide : produire un code d'erreur de sortie ou substituer des données définies aux valeurs créées par la boucle.

Recherche et gestion d'erreur

La mise au point, automatiquement activée dans LabVIEW, recherche les erreurs dans les nœuds du diagramme. Vous pouvez vérifier le paramètre de mise au point en sélectionnant **Fichier»Propriétés du VI...** et en sélectionnant **Exécution** dans le menu déroulant **Catégorie**. **Autoriser la mise au point** est coché par défaut. Chaque erreur possède un code numérique et un message d'erreur correspondant. Utilisez les VIs, les fonctions et les paramètres de gestion d'erreur pour gérer les erreurs. Par exemple, si LabVIEW rencontre une erreur, vous pouvez afficher le message d'erreur dans une boîte de dialogue. Utilisez la gestion d'erreur avec les outils de mise au point pour rechercher et gérer les erreurs. National Instruments vous recommande vivement la gestion d'erreur.

Recherche d'erreur

Même si vous êtes confiant dans les VIs que vous créez, vous ne pouvez pas prédire les problèmes que rencontreront les utilisateurs. Sans mécanisme de recherche d'erreur, vous êtes simplement au courant que le VI ne fonctionne pas correctement. La recherche d'erreur vous indique pourquoi et où l'erreur s'est produite.

Lorsque vous effectuez des opérations d'E/S, envisagez qu'une erreur peut se produire. Presque toutes les fonctions d'E/S retournent des informations sur l'erreur. Incluez la recherche d'erreur dans les VIs, particulièrement pour les opérations d'E/S (fichier, série, instrumentation, acquisition de données et communication) et fournissez un mécanisme pour gérer des erreurs de manière appropriée.

La recherche d'erreur dans les VIs peut vous aider à identifier les problèmes suivants :

- Vous avez incorrectement initialisé les communications ou vous avez écrit des données incorrectes sur un périphérique externe.
- Un périphérique n'est plus alimenté, est en panne ou fonctionne incorrectement.
- Vous avez mis à jour votre système d'exploitation, qui a changé le chemin vers un fichier, ou encore la fonctionnalité d'un VI ou de la bibliothèque. Vous pouvez remarquer un problème dans un VI ou un programme du système.

Gestion d'erreur

LabVIEW ne gère pas automatiquement les erreurs. C'est vous qui choisissez la méthode de gestion d'erreur à utiliser. Par exemple, si un VI d'E/S dans le diagramme produit un timeout, vous pouvez choisir que seule une partie de l'application s'arrête. Vous pouvez également vouloir que le VI retente son exécution pendant un certain temps. Dans LabVIEW, vous pouvez effectuer ces choix sur le diagramme du VI.

Les VIs et les fonctions retournent des erreurs de deux façons : avec des codes d'erreur numériques ou avec un cluster d'erreur. Typiquement, les fonctions utilisent des codes d'erreur numériques et les VIs utilisent un cluster d'erreur, avec en général des entrées et des sorties d'erreur. Reportez-vous à la section *Clusters d'erreur* de ce chapitre, pour plus d'informations sur les clusters d'erreur.

La gestion d'erreur dans LabVIEW suit le modèle de flux de données. Les informations d'erreur circulent dans un VI de la même façon que toutes autres données. Câblez les informations sur l'erreur du début à la fin du VI. Incluez un VI de gestion d'erreur à la fin du VI pour déterminer si le VI s'est exécuté sans produire d'erreur. Utilisez les clusters d'**entrée d'erreur** et de **sortie d'erreur** dans chaque VI que vous utilisez ou construisez pour transmettre les informations d'erreur dans le VI.

Lorsque le VI s'exécute, LabVIEW recherche les erreurs à chaque nœud d'exécution. Si LabVIEW ne trouve aucune erreur, le nœud s'exécute normalement. Si LabVIEW détecte une erreur, le nœud la transmet au nœud suivant sans exécution. Le nœud suivant fait de même, et ainsi de suite. LabVIEW reporte l'erreur à la fin de l'exécution.

Clusters d'erreur

Les clusters d'erreur, situés sur la palette **Commandes»Tableau et Cluster**, contiennent les éléments d'informations suivants :

- **état** représente une valeur booléenne qui retourne VRAI si une erreur s'est produite. La plupart des VIs, fonctions et structures qui acceptent des données booléennes reconnaissent aussi ce paramètre. Par exemple, vous pouvez câbler un cluster d'erreur aux terminaux d'entrée booléens des fonctions Arrêter, Quitter LabVIEW ou Sélectionner. Si une erreur survient, le cluster d'erreur transmet la valeur VRAI à ces fonctions.
- **code** représente un entier 32 bits signé qui identifie l'erreur numériquement. Un code d'erreur différent de zéro couplé avec un **état** FAUX signale une mise en garde plutôt qu'une erreur fatale.
- **source** représente une chaîne qui identifie l'endroit où l'erreur s'est produite.

Utilisation des boucles While pour la gestion d'erreur

Vous pouvez câbler un cluster d'erreur au terminal conditionnel d'une boucle While pour arrêter l'itération de la boucle While. Lorsque vous câblez le cluster d'erreur au terminal conditionnel, seule la valeur VRAI ou FAUX du paramètre **état** du cluster d'erreur est transmise au terminal. Lorsqu'une erreur se produit, la boucle While s'arrête.

Lorsqu'un cluster d'erreur est câblé au terminal conditionnel, les éléments du menu local **Arrêter sur condition Vraie et Continuer sur condition Vraie** deviennent **Arrêter si une erreur se produit et Continuer tant que l'erreur est présente**.

Utilisation des structures Condition pour la gestion d'erreur

Lorsque vous câblez un cluster d'erreur au terminal du sélecteur d'une structure Condition, le sélecteur de condition affiche deux conditions, **Erreur** et **Pas d'erreur**, et le cadre de la structure Condition prend la couleur rouge pour **Erreur** et verte pour **Pas d'erreur**. Lorsqu'une erreur se produit, la structure Condition exécute le sous-diagramme de la condition appropriée. Reportez-vous à la section [Structures Condition](#) du chapitre 8, [Boucles For et While et Structures Condition, Séquence et Événement](#), pour plus d'informations sur l'utilisation des structures Condition.

Création de VIs et de sous-VIs

Après avoir appris à construire une face-avant et un diagramme, vous pouvez créer vos propres VIs et sous-VIs, distribuer des VIs et construire des applications autonomes et des bibliothèques partagées.

Reportez-vous au manuel *LabVIEW Development Guidelines* pour obtenir plus d'informations sur la planification de votre projet, y compris des informations sur les difficultés courantes rencontrées lors du développement et les outils que vous pouvez utiliser pour votre projet.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour plus d'informations sur la création et l'utilisation des sous-VIs, l'enregistrement des VIs et la construction d'applications autonomes et de bibliothèques partagées.

Planification et conception de votre projet

Avant de développer vos propres VIs, créez une liste des tâches que les utilisateurs devront accomplir. Déterminez les composantes de l'interface utilisateur, le nombre et le type des commandes et des indicateurs dont vous avez besoin pour l'analyse de données, l'affichage des résultats d'analyse, etc. Réfléchissez et discutez-en avec de futurs utilisateurs ou d'autres membres de l'équipe du projet ; vous devez déterminer quand et comment l'utilisateur devra accéder aux fonctions. Créez des exemples de faces-avant pour présenter aux futurs utilisateurs ou aux membres de l'équipe du projet et déterminez si la face-avant aide vos utilisateurs à accomplir leurs tâches. Utilisez ce processus interactif pour affiner l'interface utilisateur si nécessaire.

Divisez votre application en parties faciles à gérer et dans des endroits logiques. Commencez par un diagramme de niveau supérieur qui inclut les principales composantes de votre application. Par exemple, le diagramme peut inclure un bloc pour la configuration, un bloc pour l'acquisition, un bloc pour l'analyse des données acquises, un bloc pour l'affichage des résultats d'analyse, un bloc pour enregistrer les données sur le disque et, enfin, un bloc pour la gestion d'erreur.

Après avoir conçu le diagramme de niveau supérieur, définissez les entrées et sorties. Concevez ensuite les sous-VIs constituant les principales composantes du diagramme de niveau supérieur. L'utilisation de sous-VIs rend le diagramme de niveau supérieur facile à lire, à mettre au point, à comprendre et à maintenir. Vous pouvez également créer des sous-VIs pour des opérations communes ou fréquentes, que vous pourrez réutiliser. Testez les sous-VIs pendant que vous les créez. Vous pouvez créer des routines de test de niveau supérieur, mais repérer des erreurs dans un petit module est plus facile que tester une hiérarchie de plusieurs VIs. L'utilisation de sous-VIs pour accomplir des tâches de niveau inférieur facilite la modification et la réorganisation de votre application. Reportez-vous à la section *Sous-VIs* de ce chapitre, pour plus d'informations sur les sous-VIs.

Sélectionnez **Aide»Recherche d'exemples...** pour des exemples de diagrammes et de sous-VIs.

Conception de projets avec plusieurs développeurs

Si plusieurs développeurs travaillent sur le même projet, définissez tout de suite des responsabilités pour la programmation, ainsi que des standards d'interfaces et de codage, pour assurer un processus de développement correct.

Gardez des copies maîtres des VIs du projet sur un ordinateur unique et instituez un contrôle de code source. Pensez à utiliser la version Système de développement professionnel LabVIEW, qui comporte des outils de contrôle du code source simplifiant le partage de fichiers. Ces outils comprennent aussi un utilitaire pour comparer des VIs et visualiser les changements effectués entre des versions de VIs. Reportez-vous à la section *Source Code Control* du chapitre 2, *Incorporating Quality into the Development Process*, dans le manuel *LabVIEW Development Guidelines*, pour plus d'informations sur l'utilisation du contrôle du code source.

Utilisation des VIs et fonctions intégrés

LabVIEW inclut des VIs et des fonctions pour vous aider à construire des applications spécifiques, comme les VIs et les fonctions d'acquisition de données, les VIs qui accèdent à d'autres VIs et les VIs qui communiquent avec d'autres applications. Vous pouvez utiliser ces VIs comme sous-VIs dans votre application pour réduire le temps de développement. Reportez-vous à la section *Sous-VIs* de ce chapitre, pour plus d'informations sur les sous-VIs.

Construction de VIs et de fonctions de contrôle d'instruments et d'acquisition de données

Vous pouvez utiliser les VIs et fonctions intégrés pour contrôler des instruments externes, comme des oscilloscopes, et pour acquérir des données, par exemple les mesures d'un thermocouple.

Utilisez les VIs et fonctions d'E/S d'instruments de la palette **Fonctions»E/S d'instruments** pour contrôler des instruments externes. Pour contrôler des instruments dans LabVIEW, le matériel correct doit être installé, alimenté et en bon état de marche sur votre ordinateur. Les VIs et fonctions que vous utilisez pour contrôler des instruments dépendent des protocoles de communication d'instrumentation supportés par votre matériel. Reportez-vous au *LabVIEW Measurements Manual*, pour plus d'informations sur la construction des VIs de contrôle d'instruments.

Utilisez les VIs et fonctions d'acquisition de données de la palette **Fonctions»Acquisition de données** pour acquérir des données provenant de périphériques DAQ. Pour utiliser ces VIs, le driver NI-DAQ et le matériel DAQ doivent être installés. Reportez-vous au *LabVIEW Measurements Manual* pour plus d'informations sur l'installation du driver NI-DAQ et du matériel DAQ, ainsi que sur la construction des VIs pour l'acquisition de données. Après l'acquisition de données, vous pouvez utiliser les VIs et fonctions intégrés mathématiques, d'analyse et de génération de rapport, pour réaliser des opérations mathématiques sur ces données. Reportez-vous au manuel *Analysis Concepts* pour obtenir plus d'informations sur les concepts d'analyse et mathématiques de LabVIEW.

Construction de VIs qui accèdent à d'autres VIs

Utilisez les VIs et fonctions de contrôle d'applications de la palette **Fonctions»Contrôle d'applications** pour contrôler le comportement des VIs lorsqu'ils sont appelés comme sous-VIs ou exécutés par l'utilisateur. Vous pouvez utiliser ces VIs et fonctions pour configurer plusieurs VIs à la fois. En outre, si vous êtes sur un réseau avec d'autres utilisateurs LabVIEW, vous pouvez utiliser ces VIs et fonctions pour accéder aux VIs et les contrôler à distance. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation*, pour plus d'informations sur le contrôle des VIs à distance.

Construction de VIs qui communiquent avec d'autres applications

Utilisez les VIs et fonctions d'E/S sur fichiers de la palette **Fonctions»E/S sur fichiers** pour écrire ou lire des données d'autres applications, comme par exemple Microsoft Excel. Vous pouvez utiliser ces VIs et fonctions pour générer un rapport ou incorporer des données d'une autre application dans le VI. Reportez-vous au chapitre 13, *E/S sur fichiers*, pour plus d'informations sur la transmission des données avec d'autres applications.

Utilisez les VIs et fonctions de communication de la palette **Fonctions»Communication** pour transférer des données LabVIEW sur le Web avec un protocole de communication tel que FTP et pour construire des applications client-serveur utilisant des protocoles de communication. Reportez-vous au chapitre 17, *Mise en réseau dans LabVIEW*, pour plus d'informations sur la communication avec d'autres applications sur le réseau ou sur le Web.

(Windows) Utilisez les fonctions ActiveX de la palette **Fonctions»Communication»ActiveX** pour ajouter des objets ActiveX aux VIs ou pour contrôler des applications ActiveX. Reportez-vous au chapitre 18, *ActiveX*, pour plus d'informations sur l'utilisation de la technologie ActiveX.

Sous-VIs

Lorsque vous avez construit un VI et créé une icône et un connecteur, vous pouvez l'utiliser dans un autre VI. Un VI appelé depuis le diagramme d'un autre VI se nomme un sous-VI. Un sous-VI correspond à un sous-programme dans des langages de programmation textuels. Un nœud de sous-VI correspond à un appel de sous-programme dans des langages de programmation textuels. Le nœud n'est pas le sous-VI lui-même, tout comme un appel de sous-programme dans un programme n'est pas le sous-programme lui-même. Un diagramme qui contient plusieurs nœuds de sous-VI identiques appelle le même sous-VI plusieurs fois.

Les commandes et indicateurs de sous-VI reçoivent et retournent des données avec le diagramme du VI appelant. Sélectionnez des VIs sur la palette **Fonctions»Sélection d'un VI...**, puis mettez-les sur le diagramme pour créer un appel à ces sous-VIs dans un VI.

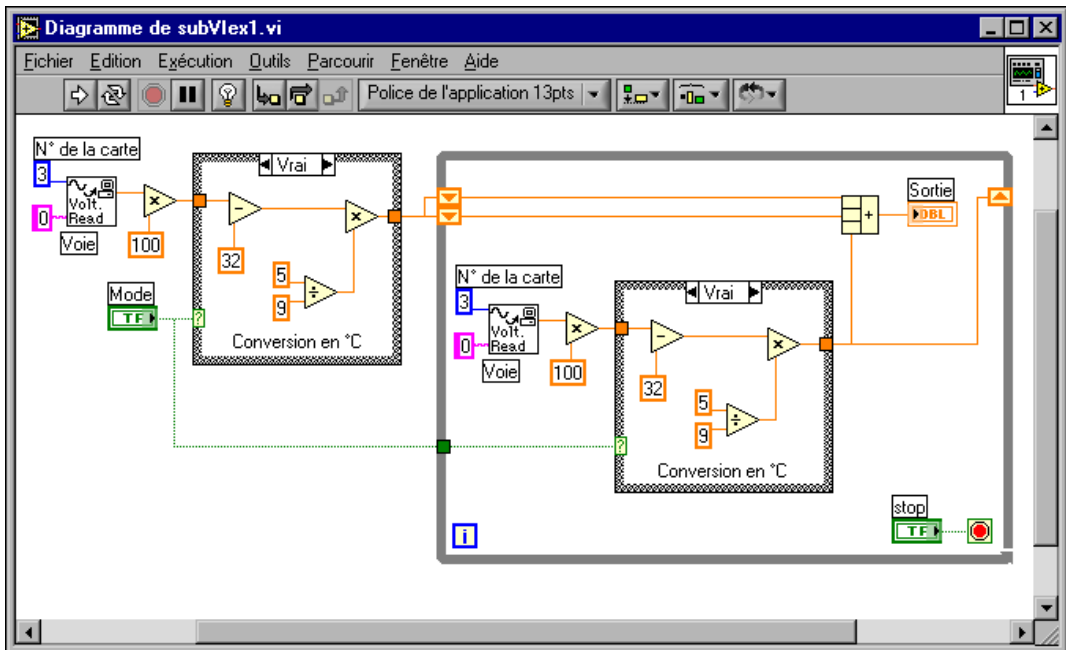


Remarque Avant de pouvoir utiliser un VI comme sous-VI, vous devez configurer un connecteur. Reportez-vous à la section *Configuration du connecteur* de ce chapitre pour plus d'informations sur la création d'un connecteur.

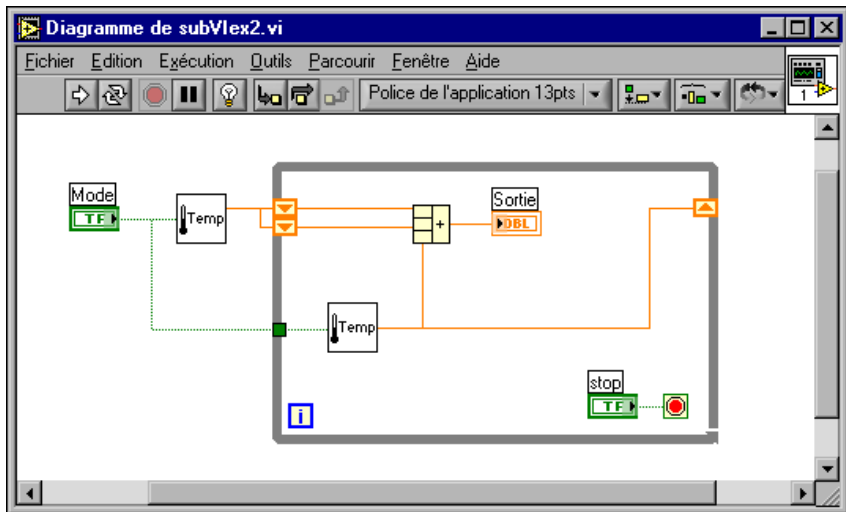
Vous pouvez éditer un sous-VI en utilisant l'outil Doigt ou l'outil Flèche pour double-cliquer sur le sous-VI du diagramme. Quand vous enregistrez le sous-VI, les modifications affectent tous les appels au sous-VI et non pas seulement l'instance actuelle.

Recherche d'opérations communes

Lorsque vous créez des VIs, vous pouvez vous rendre compte que vous accomplissez fréquemment une opération particulière. Pensez à utiliser des sous-VIs ou des boucles pour répéter cette opération. Par exemple, le diagramme suivant contient deux opérations identiques.



Vous pouvez créer un sous-VI qui effectue cette opération et appeler ce sous-VI deux fois, comme indiqué dans le diagramme suivant.



Reportez-vous au chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour plus d'informations sur l'utilisation de boucles permettant de combiner des opérations communes.

Configuration du connecteur



Pour utiliser un VI comme sous-VI, vous devez construire un connecteur, représenté sur la gauche. Le connecteur est un groupe de terminaux qui correspondent aux commandes et aux indicateurs d'un VI ; ce groupe est semblable à la liste de paramètres d'un appel de fonction dans les langages de programmation textuels. Le connecteur définit les entrées et sorties que vous pouvez connecter au VI que vous voulez utiliser comme sous-VI. Reportez-vous à la section *Icône et connecteur* du chapitre 2, *Introduction aux instruments virtuels*, pour plus d'informations sur les connecteurs.

Définissez les connexions en affectant une commande ou un indicateur de face-avant à chaque terminal du cadre connecteur. Pour définir un cadre connecteur, cliquez avec le bouton droit sur l'icône dans le coin supérieur droit de la fenêtre de la face-avant et sélectionnez **Visualiser le connecteur** dans le menu local. Le connecteur remplace l'icône. Chaque rectangle sur le connecteur représente un terminal. Utilisez les rectangles pour affecter des entrées et sorties. Le nombre de terminaux affichés par LabVIEW sur le connecteur dépend du nombre de commandes et indicateurs sur la face-avant.

Le connecteur possède 28 terminaux maximum. Si votre face-avant contient plus de 28 commandes et indicateurs que vous souhaitez utiliser, groupez certains d'entre eux dans un cluster et affectez le cluster à un terminal sur le connecteur. Reportez-vous à la section *Clusters* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour plus d'informations sur le groupement des données en utilisant des clusters.



Remarque Évitez d'attribuer plus de 16 terminaux à un VI. Un nombre trop élevé de terminaux peut réduire la capacité de lecture et d'utilisation du VI.

Sélectionnez un motif de terminaux différent pour un VI en cliquant avec le bouton droit sur le connecteur et en sélectionnant **Motifs** dans le menu local. Sélectionnez un motif de connecteur avec des terminaux supplémentaires. Vous pouvez laisser les terminaux supplémentaires non connectés tant que vous n'en avez pas besoin. Cette flexibilité vous permet d'effectuer des changements avec un effet minimal sur la hiérarchie des VIs.

Si vous créez un groupe de sous-VIs souvent utilisés ensemble, attribuez aux sous-VIs un connecteur cohérent avec les entrées communes dans le même emplacement pour vous rappeler où situer chaque entrée. Si vous créez un sous-VI qui produit une sortie utilisée par un autre sous-VI comme entrée, alignez les connexions d'entrée et de sortie pour simplifier les modèles de câblage. Placez les clusters d'**entrée d'erreur** sur le coin inférieur gauche de la face-avant et les clusters de **sortie d'erreur** sur le coin inférieur droit.

La figure 7-1 présente des exemples de clusters d'erreur mal alignés et bien alignés.

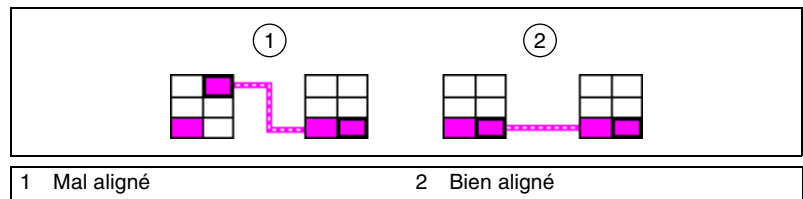


Figure 7-1. Clusters d'erreur mal alignés et bien alignés

Définition des entrées et sorties nécessaires, recommandées et optionnelles

Vous pouvez préciser quelles sont les entrées et sorties nécessaires, recommandées ou optionnelles afin que les utilisateurs n'oublient pas de câbler les connexions des sous-VIs.

Cliquez avec le bouton droit sur un terminal du connecteur et sélectionnez **Cette connexion est...** dans le menu local. Une coche indique le paramètre du terminal. Sélectionnez **Nécessaire**, **Recommandé** ou **Optionnelle**.

Le diagramme sur lequel le sous-VI est déposé sera brisé si un terminal d'entrée défini comme nécessaire n'est pas câblé. Nécessaire n'est pas une définition disponible pour les terminaux de sortie. Le diagramme sur lequel le sous-VI est déposé peut être exécuté si un terminal d'entrée ou de sortie défini comme recommandé n'est pas câblé ; cependant, la fenêtre **Liste des erreurs** rapportera une mise en garde indiquant que l'entrée n'a pas été câblée. Le diagramme sur lequel le sous-VI est déposé peut être exécuté si un terminal d'entrée ou de sortie défini comme optionnel n'est pas câblé et aucune mise en garde ne sera générée.

Les entrées et sorties des VIs dans `vi.lib` sont déjà identifiées comme **Nécessaire**, **Recommandée** ou **Optionnelle**. LabVIEW définit par défaut les entrées et sorties des VIs que vous créez comme **Recommandée**. Définissez un paramètre de terminal comme Nécessaire uniquement si l'entrée ou la sortie doit impérativement être reliée pour que le VI s'exécute correctement.

Dans la fenêtre **Aide contextuelle**, les connexions nécessaires sont en gras, les connexions recommandées sont en texte normal et les connexions optionnelles sont grisées si la vue **Détaillée** est sélectionnée ; par ailleurs, les connexions optionnelles n'apparaissent pas si la vue **Simple** est sélectionnée.

Création d'une icône



Chaque VI affiche une icône, présentée sur la gauche, dans le coin supérieur droit des fenêtres de la face-avant et du diagramme. Une icône est la représentation graphique d'un VI. Elle peut contenir du texte, des images ou une combinaison des deux. Si vous utilisez un VI comme sous-VI, l'icône identifie le sous-VI sur le diagramme du VI.

L'icône par défaut contient un nombre qui indique combien de nouveaux VIs vous avez ouverts depuis le lancement de LabVIEW. Créez des icônes personnalisées pour remplacer l'icône par défaut en cliquant avec le bouton

droit sur l'icône dans le coin supérieur droit de la face-avant ou du diagramme et en sélectionnant **Éditer l'icône** dans le menu local ou en double-cliquant sur l'icône.

Vous pouvez aussi déplacer un graphique provenant de n'importe quel endroit de votre système de fichiers et le déposer dans le coin supérieur droit de la face-avant ou du diagramme. LabVIEW convertit le graphique en une icône de 32 pixels.

Selon le type de moniteur que vous utilisez, vous pouvez concevoir une icône différente pour les modes monochrome, 16 couleurs et 256 couleurs. LabVIEW utilise l'icône monochrome pour imprimer, sauf si vous avez une imprimante couleur.

Création de sous-VIs à partir des portions d'un VI

Convertissez la portion d'un VI en sous-VI en utilisant l'outil Doigt pour sélectionner la section du diagramme que vous désirez réutiliser et en sélectionnant **Édition > Créer un sous-VI**. Une icône pour le nouveau sous-VI remplace la section du diagramme sélectionnée. LabVIEW crée des commandes et indicateurs pour le nouveau sous-VI et connecte le sous-VI aux fils de liaison existants.

Il est pratique de créer un sous-VI à partir d'une sélection, mais une planification est nécessaire pour créer une hiérarchie de VIs logique.

Conception de sous-VIs

Si les utilisateurs n'ont pas besoin d'afficher la face-avant d'un sous-VI, vous pouvez consacrer moins de temps à son apparence, y compris aux couleurs et aux polices. Cependant, l'organisation de la face-avant est toujours importante parce que vous pouvez avoir besoin de la visualiser lors de la mise au point du VI.

Placez les commandes et indicateurs comme ils apparaissent dans le connecteur. Placez les commandes sur la gauche de la face-avant et les indicateurs sur la droite. Placez les clusters d'**entrée d'erreur** sur le coin inférieur gauche de la face-avant et les clusters de **sortie d'erreur** sur le coin inférieur droit. Reportez-vous à la section [Configuration du connecteur](#) de ce chapitre pour plus d'informations sur la création d'un connecteur.

Si une commande possède une valeur par défaut, mettez la valeur par défaut entre parenthèses dans le nom de la commande. Incluez également les unités de mesure, le cas échéant, dans le nom de la commande.

Par exemple, si une commande définit la limite haute de température et a une valeur par défaut de 75 °F, appelez la commande **limite haute de température (75 deg F)**. Si vous utilisez le sous-VI sur plusieurs plates-formes, évitez d'utiliser des caractères spéciaux dans les noms de la commande. Par exemple, utilisez **deg F** au lieu de °F.

Nommez les commandes booléennes pour que les utilisateurs sachent ce que fait la commande dans l'état VRAI. Utilisez des noms comme **annuler**, **remettre à zéro** et **initialiser**, qui décrivent une action.

Affichage de la hiérarchie des VIs

La **fenêtre de hiérarchie** affiche une représentation graphique de la hiérarchie appelante de tous les VIs en mémoire, y compris les définitions de type et les variables globales. Sélectionnez **Parcourir»Visualiser la hiérarchie du VI** pour afficher la **Fenêtre de hiérarchie**. Utilisez cette fenêtre pour afficher les sous-VIs et les autres nœuds constituant le VI.

Lorsque vous déplacez l'outil Doigt sur les objets dans la **Fenêtre de hiérarchie**, LabVIEW affiche le nom de chaque VI. Vous pouvez utiliser l'outil Flèche pour déplacer un VI ou un sous-VI de la **Fenêtre de hiérarchie** vers le diagramme pour utiliser le VI ou sous-VI comme un sous-VI dans un autre VI. Vous pouvez également sélectionner et copier un ou plusieurs nœuds dans le presse-papiers et les coller sur d'autres diagrammes. Double-cliquez sur un VI dans la **Fenêtre de hiérarchie** pour afficher la face-avant de ce VI.

Un VI qui contient des sous-VIs possède un bouton flèche sur son cadre. Cliquez sur ce bouton flèche pour afficher ou masquer les sous-VIs. Un bouton flèche rouge apparaît lorsque tous les sous-VIs sont masqués. Un bouton flèche noir apparaît lorsque tous les sous-VIs sont affichés.

Enregistrement des VIs

Vous pouvez enregistrer des VIs en tant que fichiers individuels ou vous pouvez en grouper plusieurs et les enregistrer dans une bibliothèque de VIs. Les fichiers de bibliothèque de VIs se terminent par l'extension `.llb`. National Instruments vous recommande d'enregistrer les VIs comme des fichiers individuels, organisés en répertoires, particulièrement si plusieurs développeurs travaillent sur le même projet.

Avantages de l'enregistrement des VIs en tant que fichiers individuels

La liste suivante présente les raisons susceptibles de vous inciter à enregistrer les VIs en tant que fichiers individuels :

- Vous pouvez utiliser le système de fichiers pour gérer les fichiers individuels.
- Vous pouvez utiliser des sous-répertoires.
- Vous pouvez stocker des VIs et des commandes dans des fichiers individuels avec plus de sécurité que si vous stockiez votre projet entier dans le même fichier.
- Vous pouvez utiliser les outils de contrôle du code source de la version Système de développement professionnel ou d'un autre fabricant.

Avantages de l'enregistrement des VIs sous des bibliothèques

La liste suivante présente les raisons susceptibles de vous inciter à enregistrer les VIs sous des bibliothèques :

- Vous pouvez utiliser 255 caractères pour nommer vos fichiers. **(Macintosh)** MacOS 9.x ou version précédente vous limite à 31 caractères pour les noms des fichiers mais n'impose pas de limite quant au nombre de caractères pour les noms de bibliothèques.
- Vous pouvez transférer une bibliothèque de VIs sur d'autres plates-formes plus facilement que vous ne pouvez le faire avec plusieurs VIs individuels.
- Vous pouvez légèrement réduire la taille du fichier de votre projet parce que les bibliothèques de VIs sont compressées pour réduire les besoins en espace disque.
- Vous pouvez marquer les VIs contenus dans une bibliothèque comme étant de niveau supérieur ; ainsi, lorsque vous ouvrez la bibliothèque, LabVIEW ouvre automatiquement tous les VIs de niveau supérieur compris dans cette bibliothèque.



Remarque LabVIEW stocke de nombreux VIs intégrés et exemples dans des bibliothèques de VIs pour assurer des emplacements de stockage cohérents sur toutes les plates-formes.

Si vous utilisez des bibliothèques de VIs, pensez à diviser votre application en plusieurs bibliothèques de VIs. Mettez les VIs de niveau supérieur dans une bibliothèque de VIs et configurez les autres bibliothèques pour contenir des VIs suivant leur fonction. L'enregistrement des modifications apportées à un VI contenu dans une bibliothèque prend plus de temps que celui des

modifications apportées à un VI individuel car le système d'exploitation doit écrire les changements dans un fichier de taille plus importante. Enregistrer les changements apportés à une bibliothèque volumineuse peut nécessiter davantage de mémoire et réduire les performances. Essayez de limiter la taille de chaque bibliothèque à environ 1 Mo.

Gestion des VIs dans les bibliothèques

Utilisez le Gestionnaire de bibliothèques de VIs, auquel vous accédez en sélectionnant **Outils»Gestionnaire de bibliothèques de VIs...**, pour simplifier la copie, le changement de nom et la suppression de fichiers dans les bibliothèques de VIs. Vous pouvez également utiliser cet outil pour créer de nouvelles bibliothèques de VIs ou de nouveaux répertoires et pour convertir les bibliothèques de VIs en répertoires ou inversement. Ces opérations sont importantes si vous devez gérer vos VIs avec des outils de contrôle du code source.

Avant d'utiliser le Gestionnaire de bibliothèques de VIs, fermez tous les VIs qui pourraient être affectés pour éviter d'effectuer une opération de fichier sur un VI déjà en mémoire.

Appellation des VIs

Lorsque vous enregistrez des VIs, utilisez des noms descriptifs. Les noms descriptifs, comme `Moniteur de température.vi` et `Écrire et lire en série.vi`, facilitent l'identification et l'utilisation du VI. Si vous utilisez des noms ambigus, comme `VI#1.vi`, vous pouvez trouver l'identification des VIs difficile, surtout si vous avez enregistré plusieurs VIs.

Prévoyez si les utilisateurs utiliseront les VIs sur une autre plate-forme. Évitez d'utiliser des caractères que les systèmes d'exploitation réservent pour des raisons particulières, comme `\` `:` `/` `?` `*` `<` `>` et `#`.

(Macintosh) Utilisez des noms de VIs comportant moins de 31 caractères si les utilisateurs sont susceptibles d'exécuter les VIs sur MacOS 9.x ou version précédente.

Enregistrement au format antérieur

Vous pouvez enregistrer des VIs au format d'une version précédente de LabVIEW pour faciliter la mise à jour de LabVIEW et conserver les VIs dans deux versions de LabVIEW si nécessaire. Si vous mettez à jour votre version de LabVIEW, vous pouvez retourner à la dernière version des VIs.

Lorsque vous enregistrez un VI au format d'une version précédente, LabVIEW convertit non seulement ce VI mais aussi tous les autres VIs dans sa hiérarchie, les fichiers `vi.lib` exclus.

Souvent, un VI utilise une fonctionnalité qui n'était pas disponible dans la version précédente de LabVIEW. Dans ce cas, LabVIEW enregistre le VI autant que possible et produit un rapport sur ce qui n'a pas pu être converti. Le rapport apparaît immédiatement dans la boîte de dialogue **Mises en garde**. Cliquez sur le bouton **OK** pour accuser réception de ces mises en garde et fermer la boîte de dialogue. Cliquez sur le bouton **Enregistrer** pour enregistrer les mises en garde sur un fichier texte que vous pourrez consulter plus tard.

Distribution des VIs

Si vous souhaitez distribuer des VIs vers d'autres ordinateurs ou d'autres utilisateurs, décidez si vous voulez inclure le code source du diagramme pour que les utilisateurs puissent l'éditer ou si vous désirez masquer ou supprimer le diagramme. Sélectionnez **Fichier»Enregistrer avec options...** pour enregistrer les VIs sans les diagrammes pour réduire la taille des fichiers et empêcher le changement du code source par les utilisateurs. L'enregistrement d'un VI sans le diagramme empêche également les utilisateurs de déplacer le VI sur une autre plate-forme ou de convertir le VI dans une nouvelle version de LabVIEW.



Mise en garde Si vous enregistrez les VIs sans diagramme, n'écrasez *pas* les versions originales des VIs. Enregistrez les VIs dans différents répertoires ou utilisez des noms différents.

Au lieu de supprimer le diagramme, vous pouvez attribuer aux diagrammes une protection par mot de passe. Le diagramme est toujours disponible, mais les utilisateurs doivent entrer un mot de passe pour l'afficher ou l'éditer.

Une autre option pour la distribution des VIs est de créer une application autonome (un exécutable) ou une bibliothèque partagée. Une application autonome ou une bibliothèque partagée constitue un choix approprié quand les utilisateurs ne sont pas supposés éditer les VIs. Les applications autonomes contiennent des menus simplifiés.

Reportez-vous à la note d'application *Porting and Localizing LabVIEW VIs* pour obtenir de plus amples informations sur le transfert de VIs entre différentes plates-formes et la localisation des VIs.

Construction d'applications autonomes et de bibliothèques partagées

Sélectionnez **Outils»Construire une application ou une DLL...** pour utiliser l'Application Builder afin de créer des applications autonomes et des installeurs ou des bibliothèques partagées (DLL) pour des VIs. Utilisez des bibliothèques partagées si vous souhaitez appeler les VIs de la bibliothèque partagée en utilisant des langages de programmation textuels. Les bibliothèques partagées sont utiles lorsque vous souhaitez partager la fonctionnalité de VIs que vous avez construits avec des développeurs qui utilisent des langages de programmation textuels.



Remarque La version Système de développement professionnel LabVIEW inclut l'Application Builder. Si vous utilisez la version de base de LabVIEW ou la version Système de Développement complet, vous pouvez acheter l'Application Builder séparément.

Utilisez les tabulations dans la boîte de dialogue **Construire une application ou une DLL** pour configurer différents paramètres de l'application ou de la bibliothèque partagée que vous souhaitez construire. Après avoir défini ces paramètres, enregistrez-les dans un script pour pouvoir reconstruire l'application rapidement, si nécessaire.

Les utilisateurs peuvent exécuter votre application ou utiliser la bibliothèque partagée, mais ils ne peuvent pas éditer ou afficher les diagrammes.

Reportez-vous aux *Notes d'information Application Builder de LabVIEW*, pour plus d'informations sur l'installation de l'Application Builder.

Construction et édition de VIs

Cette partie décrit les caractéristiques, les VIs et les fonctions de LabVIEW que vous pouvez utiliser pour adapter le fonctionnement de vos applications à vos besoins. Les chapitres de cette section décrivent l'utilité de chaque fonctionnalité LabVIEW et donnent une description de chaque classe de VIs et de leurs fonctions.

La partie II, *Construction et édition de VIs*, comprend les chapitres suivants :

- Le chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement* décrit comment utiliser les structures du diagramme pour répéter des blocs de code et exécuter un code en fonction d'une certaine condition ou dans un ordre particulier.
- Le chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, décrit comment utiliser les chaînes, les tableaux et les clusters pour grouper des données.
- Le chapitre 10, *Variables globales et locales*, explique comment utiliser les variables globales et locales pour transférer des informations entre les lieux de votre application que vous ne pouvez pas connecter par un fil de liaison.
- Le chapitre 11, *Graphes et graphes déroulants*, explique comment utiliser les graphes et les graphes déroulants pour afficher des courbes de données sous forme graphique.
- Le chapitre 12, *VIs Graphisme et son*, explique comment afficher ou modifier des graphiques ou des sons dans des VIs.
- Le chapitre 13, *E/S sur fichiers*, explique comment effectuer des opérations d'E/S sur fichiers.
- Le chapitre 14, *Documenter et imprimer des VIs*, explique comment documenter et imprimer des VIs.

- Le chapitre 15, *Personnalisation des VIs*, explique comment configurer des VIs et des sous-VIs pour les adapter aux besoins de vos applications.
- Le chapitre 16, *Contrôle des VIs par programmation*, explique comment communiquer avec des VIs et d'autres instances de LabVIEW afin de pouvoir contrôler les VIs et LabVIEW par programmation.
- Le chapitre 17, *Mise en réseau dans LabVIEW*, explique comment utiliser les VIs pour communiquer en réseau avec d'autres processus, notamment ceux qui s'exécutent sur d'autres applications ou sur des ordinateurs à distance.
- Le chapitre 18, *ActiveX*, explique comment disposer d'un ensemble public d'objets, de commandes et de fonctions auxquels d'autres applications de Windows peuvent avoir accès.
- Le chapitre 19, *Appel de code provenant de langages de programmation textuels*, explique comment appeler des codes à partir de langages de programmation textuels et comment utiliser les DLL.
- Le chapitre 20, *Formules et équations*, explique comment utiliser des équations dans les VIs.

Boucles For et While et Structures Condition, Séquence et Événement

Les structures sont des représentations graphiques des boucles et des conditions dans les langages de programmation textuels. Utilisez les structures du diagramme pour répéter des blocs de code et exécuter un code en fonction d'une certaine condition ou dans un ordre particulier.

Comme d'autres nœuds, les structures possèdent des terminaux qui les connectent aux autres nœuds du diagramme, elles s'exécutent automatiquement lorsque des données d'entrée sont disponibles et fournissent des données aux fils de liaison de sortie lorsque l'exécution est terminée.

Chaque structure possède un cadre distinctif et redimensionnable pour contenir la section du diagramme qui s'exécute selon les règles de la structure. La section du diagramme à l'intérieur du cadre de la structure est appelée un sous-diagramme. Les terminaux qui fournissent et récupèrent les données dans les structures sont appelés des tunnels. Un tunnel est un point de connexion sur le cadre d'une structure.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour plus d'informations sur l'utilisation des structures.

Utilisez les structures suivantes de la palette **Fonctions»Structures** pour contrôler l'exécution du diagramme :

- **Boucle For** : exécute un sous-diagramme un certain nombre de fois.
- **Boucle While** : exécute un sous-diagramme jusqu'à ce qu'une condition soit remplie.
- **Structure Condition** : contient plusieurs sous-diagrammes, un seul d'entre eux s'exécutant selon la valeur d'entrée transmise à la structure.

- **Structure Séquence** : contient un ou plusieurs sous-diagrammes, qui s'exécutent séquentiellement.
- **Boîte de calcul** : réalise des opérations mathématiques sur une entrée numérique. Reportez-vous à la section *Boîtes de calcul* du chapitre 20, *Formules et équations*, pour des informations sur l'utilisation des boîtes de calcul.
- **Structure Événement** : contient un ou plusieurs sous-diagrammes s'exécutant selon l'interaction de l'utilisateur avec le VI.

Cliquez avec le bouton droit sur le cadre de la structure pour afficher le menu local.

Structures des boucles For et While

Utilisez les boucles For et While pour contrôler des opérations répétitives.

Boucles For



Une boucle For, représentée sur la gauche, exécute un sous-diagramme un certain nombre de fois.



La valeur du terminal de décompte (terminal d'entrée), représenté sur la gauche, indique combien de fois le sous-diagramme doit être répété. Définissez ce décompte explicitement en câblant une valeur provenant de l'extérieur de la boucle sur la gauche ou sur le haut du terminal de décompte, ou définissez le décompte implicitement avec l'auto-indexation. Reportez-vous à la section *Auto-indexation pour définir le décompte de la boucle For* de ce chapitre pour plus d'informations à propos du paramétrage implicite du décompte.



Le terminal d'itération (terminal de sortie), représenté sur la gauche, contient le nombre d'itérations s'étant produites. Le décompte des itérations démarre toujours à zéro. Pendant la première itération, le terminal d'itération retourne 0.

Les terminaux de décompte et d'itération sont tous les deux des entiers longs signés. Si vous câblez un nombre à virgule flottante au terminal de décompte, LabVIEW l'arrondit et le contraint dans la gamme. Si vous câblez 0 sur le terminal de décompte, la boucle ne s'exécute pas.

Ajoutez des registres à décalage à la boucle For pour transmettre des données de l'itération actuelle à l'itération suivante. Reportez-vous à la section *Registres à décalage sur des boucles* de ce chapitre pour plus d'informations sur l'ajout de registres à décalage à une boucle.

Boucles While



Comme une boucle “do” ou une boucle “repeat-until” (répéter jusqu'à...) dans les langages de programmation textuels, une boucle While, représentée sur la gauche, exécute un sous-diagramme jusqu'à ce qu'une condition soit remplie.



La boucle While exécute le sous-diagramme jusqu'à ce que le terminal conditionnel, un terminal d'entrée, reçoive une valeur booléenne spécifique. Le comportement par défaut et l'apparence du terminal conditionnel correspondent à **Continuer sur condition Vraie**, représenté sur la gauche. Lorsqu'un terminal conditionnel est réglé sur **Continuer sur condition Vraie**, la boucle While exécute son sous-diagramme jusqu'à ce que le terminal conditionnel reçoive une valeur FAUX. Vous pouvez changer le comportement et l'apparence du terminal conditionnel en cliquant avec le bouton droit sur le terminal ou sur le cadre de la boucle While et en sélectionnant **Arrêter sur condition Vraie**, présenté sur la gauche. Vous pouvez aussi utiliser l'outil Doigt pour cliquer sur le terminal conditionnel afin de changer la condition. Lorsqu'un terminal conditionnel est réglé sur **Arrêter sur condition Vraie**, la boucle While exécute son sous-diagramme jusqu'à ce que le terminal conditionnel reçoive une valeur VRAI. Comme le VI vérifie le terminal conditionnel à la fin de chaque itération, la boucle While s'exécute toujours au moins une fois. Le VI est brisé si vous ne câblez pas le terminal conditionnel.



Vous pouvez également réaliser une gestion d'erreur en utilisant le terminal conditionnel d'une boucle While. Lorsque vous câblez un cluster d'erreur au terminal conditionnel, seule la valeur VRAI ou FAUX du paramètre **état** du cluster d'erreur est transmise au terminal. Les éléments du menu local **Arrêter sur condition Vraie** et **Continuer sur condition Vraie** deviennent **Arrêter si une erreur se produit** et **Continuer tant que l'erreur est présente**. Reportez-vous à la section *Recherche et gestion d'erreur* du chapitre 6, *Exécution et mise au point des VIs*, pour plus d'informations sur la gestion et les clusters d'erreur.



Le terminal d'itération (terminal de sortie), représenté sur la gauche, contient le nombre d'itérations s'étant produites. Le décompte des itérations démarre toujours à zéro. Pendant la première itération, le terminal d'itération retourne 0.

Ajoutez des registres à décalage à la boucle While pour transmettre des données de l'itération actuelle à l'itération suivante. Reportez-vous à la section *Registres à décalage sur des boucles* de ce chapitre pour plus d'informations sur l'ajout de registres à décalage à une boucle.

Éviter des boucles While infinies

Si vous positionnez le terminal de la commande booléenne à l'extérieur de la boucle While, comme l'illustre la figure 8-1, que la commande est définie sur FAUX et que le terminal conditionnel est défini à **Arrêter sur condition Vraie** lorsque la boucle démarre, vous créez une boucle infinie. Vous créez également une boucle infinie si la commande à l'extérieur de la boucle est égale à VRAI et si le terminal conditionnel est défini sur **Continuer sur condition Vraie**.

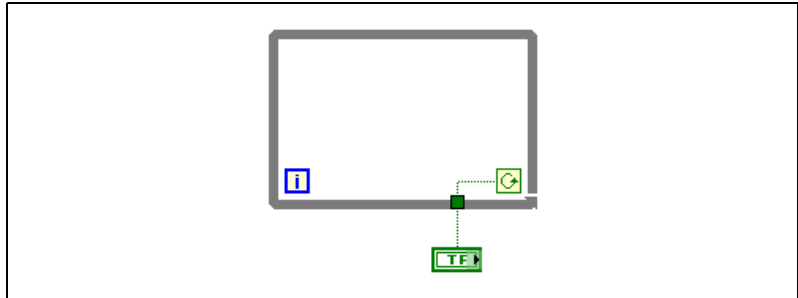


Figure 8-1. Boucle While infinie

Le changement de la valeur de la commande n'arrête pas la boucle infinie parce que la valeur n'est lue qu'une fois, avant que la boucle ne démarre. Pour arrêter une boucle infinie, vous devez stopper le VI en cliquant sur le bouton **Abandonner l'exécution** de la barre d'outils.

Auto-indexation des boucles

Si vous câblez un tableau à une boucle For ou une boucle While, vous pouvez lire et traiter chaque élément de ce tableau en activant l'auto-indexation. Reportez-vous au chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters* pour plus d'informations sur les tableaux.

Lorsque vous câblez un tableau à un tunnel d'entrée situé sur le cadre de la boucle et que vous activez l'auto-indexation sur le tunnel d'entrée, les éléments de ce tableau entrent dans la boucle au rythme d'un élément par itération, en commençant par le premier élément. Lorsque l'auto-indexation est désactivée, le tableau entier est transmis dans la boucle. Lorsque vous auto-indexez un tunnel de sortie, le tableau de sortie qui y est câblé reçoit un nouvel élément à chaque itération de la boucle. Ainsi, les tableaux de sortie auto-indexés ont toujours une taille égale au nombre d'itérations.

Cliquez avec le bouton droit sur le tunnel au bord de la boucle et sélectionnez **Activer l'indexation** ou **Désactiver l'indexation** dans le menu local pour activer ou désactiver l'auto-indexation. L'auto-indexation pour les boucles While est désactivée par défaut.

La boucle indexe les éléments scalaires à partir des tableaux 1D, les tableaux 1D à partir des tableaux 2D, etc. L'opposé se produit sur les tunnels de sortie. Les éléments scalaires s'accumulent séquentiellement dans des tableaux 1D, les tableaux 1D s'accumulent dans des tableaux 2D, etc.

Auto-indexation pour définir le décompte de la boucle For

Si vous activez l'auto-indexation sur un tableau câblé à une boucle For, vous n'avez pas besoin de câbler explicitement le terminal de décompte. Dans la mesure où vous pouvez utiliser des boucles For pour traiter les éléments d'un tableau un par un, LabVIEW active l'auto-indexation par défaut pour chaque tableau que vous câblez à une boucle For. Désactivez l'auto-indexation si vous n'avez pas besoin de traiter des tableaux élément par élément.

Si vous activez l'auto-indexation pour plusieurs tunnels ou si vous câblez également le terminal de décompte, le décompte sera le plus petit parmi les choix. Par exemple, si deux tableaux auto-indexés entrent dans la boucle, avec respectivement 10 et 20 éléments et si vous câblez une valeur de 15 sur le terminal de décompte, la boucle s'exécute 10 fois et n'indexe que les 10 premiers éléments du deuxième tableau. Si vous tracez des données provenant de deux sources sur un graphe et que vous voulez positionner les 100 premiers éléments, câblez 100 sur le terminal de décompte. Si l'une des sources de données ne comporte que 50 éléments, la boucle s'exécute 50 fois et indexe uniquement les 50 premiers éléments. Utilisez la fonction Taille d'un tableau pour déterminer la taille des tableaux.

Lorsque vous auto-indexez un tunnel de sortie d'un tableau, le tableau de sortie reçoit un nouvel élément à chaque itération de la boucle. Ainsi, les tableaux de sortie auto-indexés ont toujours une taille égale au nombre d'itérations. Par exemple, si la boucle s'exécute 10 fois, le tableau de sortie possède 10 éléments. Si vous désactivez l'auto-indexation sur un tunnel de sortie, seul l'élément de la dernière itération de la boucle est transmis au nœud suivant dans le diagramme. Un symbole entre crochets apparaît sur le cadre de la boucle pour indiquer que l'auto-indexation est activée. L'épaisseur du fil de liaison entre le tunnel de sortie et le nœud suivant indique également si la boucle utilise l'auto-indexation. Le fil de liaison est

plus épais si vous utilisez l'auto-indexation parce qu'il contient un tableau, au lieu d'un scalaire.

Auto-indexation avec des boucles While

Si vous activez l'auto-indexation pour un tableau entrant dans une boucle While, cette dernière indexe le tableau au même titre qu'une boucle For. Cependant, le nombre d'itérations exécutées par une boucle While n'est pas limité par la taille du tableau parce que la boucle While continue l'itération jusqu'à ce qu'une condition spécifique soit remplie. Lorsqu'une boucle While procède à une indexation au-delà de la fin du tableau d'entrée, la valeur par défaut du type d'élément de tableau est transmis à la boucle. Vous pouvez empêcher que la valeur par défaut soit transmise à la boucle While en utilisant la fonction Taille d'un tableau. La fonction Taille d'un tableau indique combien d'éléments sont dans le tableau. Réglez la boucle While pour qu'elle arrête son exécution après un nombre d'itérations égal à la taille du tableau.



Mise en garde Comme vous ne pouvez pas déterminer à l'avance la taille du tableau de sortie, l'activation de l'auto-indexation pour la sortie d'une boucle For est plus efficace qu'avec une boucle While. De trop nombreuses itérations peuvent épuiser la mémoire de votre système.

Registres à décalage sur des boucles

Utilisez les registres à décalage sur les boucles For et While pour transférer des valeurs d'une itération de la boucle à la suivante. Les registres à décalage sont similaires aux variables statiques des langages de programmation textuels.



Un registre à décalage apparaît sous la forme d'une paire de terminaux, représentés sur la gauche, directement à l'opposé l'un de l'autre sur les côtés verticaux du cadre de la boucle. Le terminal de droite contient une flèche vers le haut et stocke les données à la fin d'une itération. LabVIEW transfère les données connectées à droite du registre jusqu'à l'itération suivante. Créez un registre à décalage en cliquant avec le bouton droit sur le bord gauche ou droit d'une boucle et en sélectionnant **Ajouter un registre à décalage** dans le menu local.

Un registre à décalage transfère tout type de données et les change automatiquement dans le type de données du premier objet câblé au registre à décalage. Les données câblées aux terminaux de chaque registre à décalage doivent être du même type. Vous pouvez créer plusieurs registres à décalage sur une structure et vous pouvez avoir plusieurs terminaux à gauche pour vous rappeler plusieurs valeurs précédentes.

Après l'exécution de la boucle, la dernière valeur stockée dans le registre à décalage reste dans le terminal droit. Si vous câblez le terminal à l'extérieur de la boucle, le fil de liaison transfère la dernière valeur stockée dans le registre à décalage.

Si vous n'initialisez pas le registre, la boucle utilise la valeur écrite sur le registre lors de sa dernière exécution ou la valeur par défaut du type de données si la boucle n'a jamais été exécutée.

Utilisez une boucle avec un registre à décalage non initialisé pour exécuter un VI de façon répétitive de sorte qu'à chaque exécution du VI la sortie initiale du registre à décalage soit égale à la dernière valeur de l'exécution précédente. Utilisez un registre à décalage non initialisé pour conserver les informations d'état entre les exécutions ultérieures d'un VI.

Contrôle de la synchronisation

Vous pouvez vouloir contrôler la vitesse d'exécution, par exemple la vitesse à laquelle les données sont positionnées sur un diagramme. Vous pouvez utiliser la fonction Attendre dans la boucle pour attendre pendant une durée voulue exprimée en millisecondes avant une nouvelle exécution de la boucle.

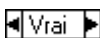
Structures Condition, Séquence et Événement

Les structures Condition, Séquence et Événement contiennent plusieurs sous-diagrammes, un seul d'entre eux étant visible à la fois. Une structure Condition ou Événement exécute un sous-diagramme selon la valeur d'entrée transmise à la structure. Une structure Séquence exécute tous ses sous-diagrammes dans un ordre donné.

Structures Condition



Une structure Condition, représentée sur la gauche, possède au moins deux sous-diagrammes ou conditions. Un seul sous-diagramme est visible à la fois et la structure exécute seulement une condition à la fois. Une valeur d'entrée détermine quel sous-diagramme s'exécute. La structure Condition est similaire aux "if...then...else" des langages de programmation textuels.



Le sélecteur de condition, en haut de la structure Condition, présenté sur la gauche, contient la condition au centre et les boutons d'incrémentement et de décrémentation de chaque côté. Utilisez les flèches d'incrémentement et de décrémentation pour faire défiler les conditions disponibles. Vous pouvez

également cliquer sur la flèche descendante située à côté de la condition et sélectionner une condition à partir du menu déroulant.



Câblez une valeur d'entrée ou un sélecteur au terminal de sélection, représenté sur la gauche, pour déterminer la condition qui s'exécute. Vous devez câbler un entier, une valeur booléenne, une chaîne ou une valeur d'énumération sur le terminal de sélection. Vous pouvez positionner le terminal de sélection n'importe où sur le bord gauche de la structure Condition. Si le terminal de sélection est de type booléen, la structure possède une condition VRAI et une condition FAUX. Si le terminal de sélection est un entier, une chaîne ou une valeur de type énumération, la structure peut avoir un nombre arbitraire de conditions.

Spécifiez une condition par défaut pour la structure Condition afin de gérer les valeurs hors gamme. Sinon, vous devez lister de façon explicite toutes les valeurs d'entrée possibles. Par exemple, si le sélecteur est un entier et que vous spécifiez des conditions pour les valeurs 1, 2 et 3, vous devez spécifier une condition par défaut à exécuter lorsque la valeur d'entrée est 4 ou tout autre valeur entière valide mais différente de 1, 2 et 3.

Valeurs du sélecteur de condition et types de données

Vous pouvez entrer une valeur unique ou des listes et des gammes de valeurs dans le sélecteur de condition. Pour des listes, séparez les valeurs avec des virgules. Pour des valeurs numériques, spécifiez une gamme ainsi : 10 . . 20, ce qui signifie tous les nombres de 10 à 20 inclus. Vous pouvez également utiliser des gammes ouvertes. Par exemple, . . 100 représente tous les nombres inférieurs ou égaux à 100. Vous pouvez également combiner des listes et des gammes, par exemple . . 5, 6, 7 . . 10, 12, 13, 14. Lorsque vous entrez des valeurs ayant des gammes qui se chevauchent dans le même sélecteur de condition, la structure Condition affiche à nouveau le sélecteur dans un format plus compact. L'exemple précédent est réaffiché ainsi : . . 10, 12 . . 14. Pour les gammes de chaînes de caractères, une gamme de a . . c inclut a et b, mais pas c. La gamme a . . c, c inclut la valeur finale c.

Lorsque vous entrez des chaînes et des énumérations dans un sélecteur de condition, les valeurs s'affichent entre guillemets, par exemple **"rouge"**, **"vert"** et **"bleu"**. Cependant, vous n'avez pas besoin de taper les guillemets lorsque vous entrez les valeurs, sauf si la chaîne ou l'énumération contient une virgule ou un symbole de gamme (" , " ou " . . "). Dans une valeur de type chaîne, utilisez des codes spéciaux de barre oblique inverse pour les caractères non alphanumériques, comme \r pour

un retour chariot, \n pour un retour à la ligne et \t pour une tabulation. Reportez-vous à l'*Aide LabVIEW* pour une liste de ces codes de barre oblique inverse.

Si vous changez le type de données du fil de liaison connecté au terminal de sélection d'une structure Condition, cette dernière convertit automatiquement les valeurs du sélecteur de condition en un nouveau type de données, lorsque c'est possible. Si vous convertissez une valeur numérique, par exemple 19, en une chaîne, la valeur de la chaîne est égale à "19". Si vous convertissez une chaîne en une valeur numérique, LabVIEW convertit seulement les valeurs de chaînes qui représentent un nombre. Les autres valeurs restent des chaînes. Si vous convertissez un nombre en une valeur booléenne, LabVIEW convertit 0 en FAUX et 1 en VRAI et toutes les autres valeurs numériques deviennent des chaînes.

Si vous entrez une valeur de sélecteur qui n'est pas du même type que l'objet câblé au terminal du sélecteur, la valeur apparaît en rouge pour indiquer que vous devez supprimer ou éditer la valeur avant que la structure ne puisse s'exécuter et le VI ne s'exécutera pas. De même, en raison de l'erreur d'arrondi possible inhérente à l'arithmétique, vous ne pouvez pas utiliser des valeurs numériques à virgule flottante comme valeurs du sélecteur de condition. Si vous câblez une valeur à virgule flottante à la condition, LabVIEW arrondit la valeur à l'entier pair le plus proche. Si vous tapez une valeur à virgule flottante dans le sélecteur de condition, la valeur apparaît en rouge pour indiquer que vous devez supprimer ou éditer la valeur avant de pouvoir exécuter la structure.

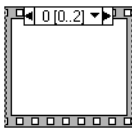
Tunnels d'entrée et de sortie

Vous pouvez créer plusieurs tunnels d'entrée et de sortie pour une structure Condition. Les entrées sont disponibles pour toutes les conditions, mais ces dernières n'ont pas besoin d'utiliser chaque entrée. Cependant, vous devez définir chaque tunnel de sortie pour chaque condition. Lorsque vous créez un tunnel de sortie dans une condition, les tunnels apparaissent à la même position sur le cadre dans toutes les autres conditions. Si au moins un tunnel de sortie n'est pas câblé, tous les tunnels de sortie sur la structure apparaissent comme des carrés blancs. Vous pouvez définir une source de données différente pour le même tunnel de sortie dans chaque condition, mais les types de données doivent être compatibles. Vous pouvez également effectuer un clic droit sur le tunnel de sortie et sélectionner **Utiliser la valeur par défaut si non câblé** à partir du menu local pour utiliser la valeur par défaut définie pour tous les tunnels non câblés.

Utilisation des structures Condition pour la gestion d'erreur

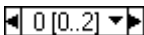
Lorsque vous câblez un cluster d'erreur au terminal de sélection d'une structure Condition, le sélecteur de condition affiche deux conditions, Erreur et Pas d'erreur, et le cadre de la structure Condition prend la couleur rouge pour Erreur et verte pour Pas d'erreur. La structure Condition exécute le sous-diagramme approprié suivant l'état de l'erreur. Reportez-vous à la section *Gestion d'erreur* du chapitre 6, *Exécution et mise au point des VIs* pour plus d'informations sur la gestion des erreurs.

Structures Séquence



Une structure Séquence, représentée sur la gauche, contient un ou plusieurs sous-diagrammes ou cadres, qui s'exécutent dans un ordre séquentiel. L'étiquette du cadre située au sommet de la structure Séquence est similaire à celle du sélecteur de condition de la structure Condition. L'étiquette du cadre contient, en son centre, le numéro du cadre et, de chaque côté, des flèches d'incrément et de décrémentation. Utilisez ces flèches pour faire défiler les cadres disponibles. Vous pouvez également cliquer sur la flèche descendante située à côté du numéro du cadre et sélectionner un cadre à partir du menu déroulant. Contrairement au sélecteur de condition, vous ne pouvez pas entrer de valeurs dans l'étiquette du cadre. Quand vous ajoutez, supprimez ou réarrangez des cadres dans une structure Séquence, LabVIEW ajuste automatiquement les nombres situés dans les étiquettes de cadres.

Une structure Séquence exécute le cadre 0, le cadre 1, le cadre 2 et ce, jusqu'au dernier cadre. La structure Séquence ne termine pas son exécution et ne retourne aucune donnée tant que le dernier cadre n'a pas terminé de s'exécuter.



Le sélecteur de séquence situé en haut de la structure Séquence, présenté sur la gauche, contient le numéro du cadre actuel et la gamme de cadres en son centre, ainsi que des boutons d'incrément et de décrémentation sur chacun de ses côtés. Par exemple, dans le sélecteur de séquence présenté à gauche, 0 est le numéro du cadre actuel et [0..2] représente la gamme de cadres. Utilisez les boutons d'incrément et de décrémentation pour faire défiler les cadres disponibles.

Utilisez la structure Séquence pour contrôler l'ordre d'exécution lorsqu'il n'existe pas de dépendance naturelle des données. Un nœud qui reçoit des données d'un autre nœud dépend du premier nœud quant aux données et s'exécute toujours après que le premier nœud a terminé son exécution.

Dans chaque cadre d'une structure Séquence, comme dans le reste du diagramme, la dépendance des données détermine l'ordre d'exécution des nœuds. Reportez-vous à la section *Dépendance des données et dépendance des données artificielle* du chapitre 5, *Construction du diagramme*, pour plus d'informations sur la dépendance des données.

Les tunnels des structures Séquence ne peuvent avoir qu'une source de données, à la différence des structures Condition. La sortie peut émettre de n'importe quel cadre, mais les données quittent la structure Séquence uniquement lorsque tous les cadres ont terminé leur exécution et non quand les cadres ont individuellement terminé leur exécution. Comme pour les structures Condition, les données câblées aux tunnels d'entrée sont disponibles pour tous les cadres.



Pour transmettre des données d'un cadre à un autre, utilisez un terminal de variable locale de séquence, représenté sur la gauche. Une flèche pointant vers l'extérieur apparaît dans le terminal de variable locale de séquence sur le cadre qui contient la source de données. Le terminal dans les cadres ultérieurs contient une flèche pointant vers l'intérieur, indiquant que le terminal est une source de données pour ce cadre. Vous ne pouvez pas utiliser le terminal de variable locale de séquence dans des cadres se trouvant avant le premier cadre dans lequel vous avez câblé la variable locale de séquence.

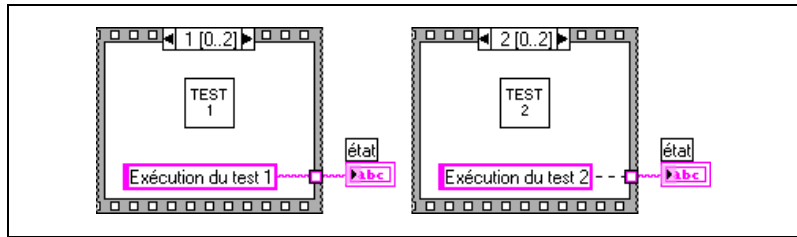
Reportez-vous à `exemples\general\structs.llb` pour des exemples d'utilisation des structures Séquence.

Éviter un usage excessif des structures Séquence

Pour profiter du parallélisme inhérent à LabVIEW, évitez l'usage excessif des structures Séquence. Les structures Séquence garantissent l'ordre d'exécution et interdisent les opérations en parallèle. Par exemple, les tâches asynchrones qui utilisent des périphériques d'E/S, comme les périphériques PXI, GPIB, ports série et DAQ, peuvent s'exécuter en même temps que d'autres opérations si les structures Séquence ne les en empêchent pas. Les structures Séquence masquent également des sections du diagramme et interrompent le flux naturel des données de gauche à droite.

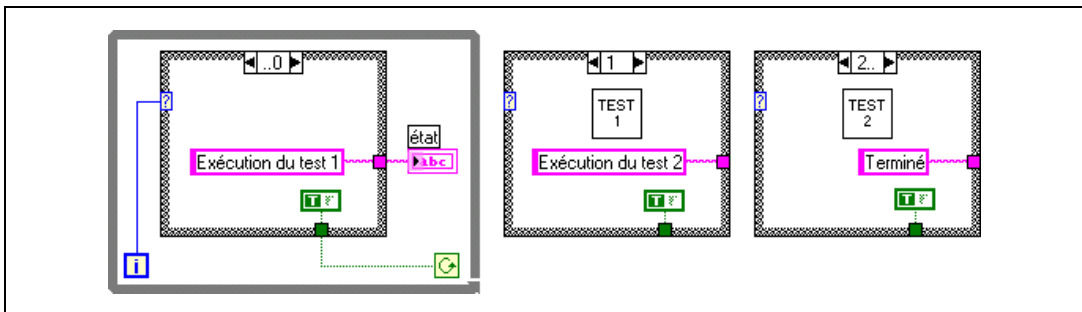
Lorsque vous devez contrôler l'ordre d'exécution, pensez à établir une dépendance des données entre les nœuds. Par exemple, vous pouvez utiliser l'E/S d'erreurs pour contrôler l'ordre d'exécution des E/S. Reportez-vous à la section *Gestion d'erreur* du chapitre 6, *Exécution et mise au point des VIs*, pour plus d'informations sur l'E/S d'erreurs.

Par ailleurs, n'utilisez pas de structures Séquence si vous voulez mettre un indicateur à jour à partir de différents cadres d'une structure. Par exemple, un VI utilisé dans une application de tests peut avoir un indicateur **état** qui affiche le nom du test en cours. Si chaque test est un sous-VI appelé à partir d'un cadre différent, vous ne pouvez pas mettre l'indicateur à jour à partir de chaque cadre, comme montré par le fil de liaison brisé dans le diagramme suivant :



Étant donné que tous les cadres de la structure Séquence s'exécutent avant toute transmission de données, un seul cadre peut attribuer une valeur à l'indicateur **état**.

Utilisez à la place une structure Condition et une boucle While, comme montré dans l'exemple suivant.



Chaque condition de la structure Condition est équivalent à un cadre de la structure Séquence. Chaque itération de la boucle While exécute la condition suivante. L'indicateur **état** affiche l'état du VI pour chaque condition. L'indicateur **état** est mis à jour dans la condition précédant celle qui appelle le sous-VI correspondant car les valeurs sont transmises en dehors de la structure après l'exécution de chaque condition.

Contrairement à une structure Séquence, une structure Condition peut transmettre des données pour terminer la boucle While au cours de n'importe quelle condition. Par exemple, si une erreur se produit au cours

du premier test, la structure Condition peut transmettre FAUX au terminal conditionnel afin de terminer la boucle. Cependant, une structure Séquence doit exécuter tous ses cadres, même si une erreur se produit.

Programmation événementielle

LabVIEW est un environnement de programmation où le flux de données détermine l'ordre d'exécution des éléments du diagramme. Avec les événements disponibles dans LabVIEW, l'interaction directe de l'utilisateur avec des objets de la face-avant et l'ordre des éléments dans le diagramme peuvent déterminer le flux de données.



Remarque La programmation événementielle est uniquement disponible dans les versions systèmes de développement complet et professionnel de LabVIEW.

Que sont les événements ?

Les événements sont causés par des actions de l'utilisateur. Par exemple, cliquer avec la souris génère un événement de souris, appuyer sur une touche du clavier génère un événement de clavier, etc. Dans un programme contrôlé par des événements, le programme doit d'abord attendre que les événements aient lieu, répondre ensuite à ces événements et enfin, attendre les événements suivants. La façon dont le programme répond dépend du code écrit pour cet événement particulier. L'ordre dans lequel le programme contrôlé par des événements s'exécute dépend des événements qui ont lieu et de l'ordre dans lequel ils se produisent.

Utilisation des événements avec LabVIEW

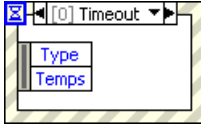
Les événements sont très utiles pour la conception d'interfaces utilisateur. Utilisez des événements pour synchroniser vos actions sur la face-avant avec l'exécution du diagramme. Vous pouvez utiliser des événements pour détecter quand un utilisateur change une valeur, ferme la face-avant, quitte une application, etc. En utilisant une structure Événement pour exécuter du code spécifique à un événement, vous réduisez le besoin qu'a le diagramme de questionner la face-avant pour déterminer les actions de l'utilisateur. Ceci diminue le temps d'exécution et simplifie le diagramme.



Remarque Les événements attendent uniquement une action directe de l'utilisateur sur la face-avant. Les événements ne fonctionnent pas si vous utilisez le VI Serveur, des variables globales ou locales et ainsi de suite pour modifier des VIs ou des objets de face-avant par programmation.

Reportez-vous à l'*Aide LabVIEW* pour plus d'informations sur les mises en garde concernant l'utilisation de structures Événement et sur les événements disponibles.

Structures Événement



Utilisez la structure Événement, ci-contre à gauche, pour gérer des événements dans une application. Comme avec une structure Condition, vous pouvez ajouter plusieurs sous-diagrammes ou conditions à la structure Événement. Vous pouvez alors configurer ces conditions pour qu'elles gèrent un ou plusieurs événements. Quand ces événements ont lieu, LabVIEW exécute la condition correspondante. Placez des structures Événement à l'intérieur de boucles pour gérer une série d'événements jusqu'à ce qu'une condition de fin d'exécution soit rencontrée.



Mise en garde Si vous placez une structure Événement à l'intérieur d'une boucle While et qu'une condition de fin d'exécution arrête la boucle, les événements que vous pourriez générer par la suite ne seront pas traités par la structure Événement car celle-ci ne s'exécutera pas. Ceci peut entraîner le verrouillage de votre interface utilisateur.



Remarque Si vous construisez un VI qui contient une structure Événement en utilisant les versions Système de développement professionnel ou complet de LabVIEW, vous pouvez exécuter ce VI dans la version de base de LabVIEW, qui conserve alors la fonctionnalité de la structure Événement. Cependant, vous ne pouvez pas éditer la structure Événement dans la version de base de LabVIEW.



Utilisez le terminal de timeout situé dans le coin supérieur gauche de la structure Événement, ci-contre, pour spécifier le nombre de millisecondes pendant lesquelles une structure Événement attend qu'un événement se produise. La valeur par défaut est -1, ce qui signifie un temps d'attente illimité. Si le timeout expire avant qu'un événement ne se produise et que vous avez configuré une condition pour gérer l'événement de type timeout, LabVIEW générera un événement de type timeout.



Le nœud de données d'événement, ci-contre, possède une apparence similaire à celle de la fonction Désassembler par nom. Ce nœud est attaché aux bords intérieurs gauche et droit de chaque condition d'événement. Vous pouvez redimensionner ce nœud verticalement et définir pour chacun de ses éléments l'accès à un champ de données d'événement arbitraire, tout comme le permet la fonction Désassembler par nom. Le nœud affiche des données qui sont différentes pour chaque condition de la structure Événement et qui dépendent de l'(des) événement(s) géré(s) par la condition que vous avez configurée. Si vous configurez une condition

unique pour gérer des événements multiples, seules les données qui sont communes à tous les types d'événements gérés par la condition sont disponibles.



Le sélecteur d'événement situé au sommet de la structure Événement, ci-contre, indique la condition actuellement sélectionnée, y compris les événements gérés par cette condition.



Remarque Tout comme la structure Condition, la structure Événement supporte les tunnels. Cependant, par défaut, vous ne devez pas câbler les tunnels de sortie de la structure Événement pour chaque condition. Tous les tunnels non câblés utilisent la valeur par défaut pour le type de données du tunnel. Effectuez un clic droit sur un tunnel et désélectionnez **Utiliser la valeur par défaut si non câblé** à partir du menu local pour revenir au comportement par défaut de la structure Condition pour laquelle les tunnels doivent toujours être câblés.

Les événements sont groupés selon des classes, telles que Application, VI et Commandes. Les données d'événement comprennent toujours une référence à l'objet qui a déclenché l'événement. Si une condition unique gère plusieurs événements associés à des commandes de classes de VI Serveur différentes, la référence à la commande de source d'événement est la classe parente commune à toutes les classes. Par exemple, si vous configurez une condition unique dans la structure Événement pour gérer des événements associés à une commande numérique et à une commande rampe de couleurs, le type de la référence à la commande de source d'événement est numérique car les commandes numérique et rampe de couleurs appartiennent toutes deux à la classe numérique.

Configuration des événements

Effectuer un clic droit sur le bord de la structure Événement et sélectionnez **Editer les événements gérés par cette condition** dans le menu local afin d'afficher la boîte de dialogue **Éditer les événements**. Utilisez cette boîte de dialogue pour éditer une ou plusieurs conditions.

Événements de type filtre ou notification

Il y a deux types d'événements : notification et filtre. Les événements de type notification informent LabVIEW qu'une action de l'utilisateur a déjà eu lieu, comme quand un utilisateur change la valeur d'une commande. Par exemple, vous pouvez écrire un VI qui notifie la structure Événement du diagramme lorsque l'utilisateur clique sur un bouton de la face-avant. Du fait que la valeur change quand l'utilisateur clique sur le bouton, la structure Événement en est informée. La structure Événement peut gérer

plusieurs événements de type notification à la fois. Si vous configurez une condition unique pour gérer plusieurs événements de type notification, seules les données qui sont communes à tous les types d'événements gérés sont disponibles.

Les événements de type filtre permettent de contrôler la façon dont l'interface utilisateur se comporte. Avec des événements de type filtre, vous pouvez valider ou modifier les données d'événement avant que l'interface utilisateur ne les traite, ou vous pouvez les éliminer complètement, empêchant ainsi tout changement d'affecter le VI. Par exemple, vous pouvez configurer un événement de type Sélection dans un menu qui n'autorise pas l'utilisateur à fermer la face-avant d'un VI de façon interactive. Vous pouvez aussi configurer un événement de type Touche enfoncée pour modifier tous les caractères tapés dans une commande chaîne afin qu'ils soient en majuscules ou encore, vous pouvez modifier l'événement pour filtrer des caractères non souhaités.

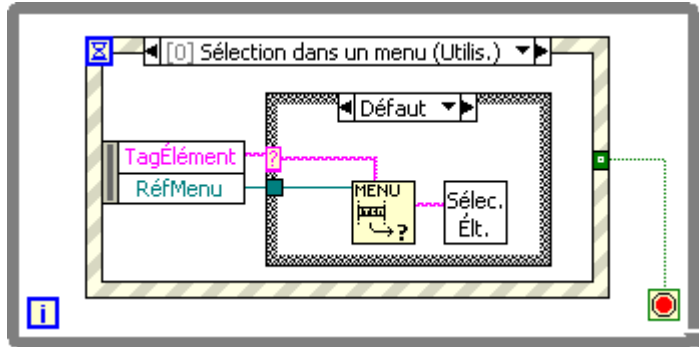
L'existence de terminaux sur le côté droit de la structure Événement, tels que le terminal Annuler ?, identifie un événement de type filtre.



Remarque Contrairement aux événements de type notification, une condition unique de la structure Événement ne peut gérer plusieurs événements de type filtre que si les données retournées par les événements sont identiques. Si vous essayez de grouper deux événements qui ne retournent pas des données identiques, tels que Touche enfoncée et Clic de souris, le VI ne s'exécute pas.

Exemple d'événement

L'exemple suivant montre une structure Événement configurée avec l'événement Sélection dans un menu (Utilis.). Cet événement capture des sélections de menu effectuées en utilisant le menu défini par l'utilisateur et nommé `sample.rtm`. **TagÉlément** retourne l'élément de menu qui a été sélectionné et **RéfMenu** retourne la référence (refnum) à la barre de menus. Reportez-vous à la bibliothèque `exemples\general\uievents.llb` pour consulter davantage d'exemples d'utilisation d'événements.



Remarque Si vous configurez la structure Événement pour qu'elle gère des événements de menu et que vous avez configuré la fonction Élément de menu sélectionné pour le même élément de menu, la structure Événement prédomine et LabVIEW ignore la fonction Élément de menu sélectionné. Quel que soit le VI, vous devriez cependant utiliser soit la structure Événement, soit la fonction Élément de menu sélectionné.

Groupement des données au moyen de chaînes, de tableaux et de clusters

Utilisez des chaînes, des tableaux et des clusters pour grouper des données. Les chaînes regroupent les séquences de caractères ASCII. Les tableaux regroupent les éléments de données du même type. Les clusters regroupent les éléments de données de types différents.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour plus d'informations sur le groupement des données en utilisant des chaînes, des tableaux et des clusters.

Chaînes

Une chaîne est une séquence de caractères ASCII affichables ou non. Les chaînes fournissent un format indépendant de la plate-forme pour les informations et les données. Certaines des applications les plus courantes pour les chaînes offrent les fonctions suivantes :

- Création de messages de texte simple.
- Transmission de données numériques sous forme de chaînes à des instruments puis conversion des chaînes en valeurs numériques.
- Stockage des données numériques sur le disque. Pour stocker les valeurs numériques dans un fichier ASCII, vous devez d'abord convertir les valeurs numériques en chaînes avant d'écrire les valeurs numériques sur un fichier du disque.
- Notification de l'utilisateur par des messages ou des instructions avec des boîtes de dialogue.

Sur la face-avant, les chaînes apparaissent comme des tables, des boîtes d'entrée de texte et des étiquettes. Éditez et manipulez les chaînes avec les fonctions Chaîne sur le diagramme. Formatez les chaînes pour pouvoir les

utiliser dans d'autres applications, comme des applications de traitement de texte et des tableurs ou dans d'autres VIs et fonctions.

Chaînes sur la face-avant

Utilisez la commande et l'indicateur de chaîne, sur la palette **Commandes» Chaîne & chemin**, pour simuler des boîtes d'entrée de texte et des étiquettes. Reportez-vous à la section *Commandes et indicateurs de type chaîne* du chapitre 4, *Construction de la face-avant*, pour plus d'informations sur les commandes et indicateurs de chaîne.

Types d'affichage de chaîne

Cliquez avec le bouton droit sur une commande ou un indicateur de chaîne sur la face-avant pour sélectionner les types d'affichage présentés dans la table 9-1. La table présente également un exemple de message dans chaque type d'affichage.

Tableau 9-1. Types d'affichage de chaîne

Type d'affichage	Description	Message
Affichage normal	Affiche des caractères imprimables en utilisant la police de la commande. Les caractères non imprimables s'affichent sous forme de boîtes.	Il existe quatre types d'affichage. \ est une barre oblique inverse.
Affichage des codes '\'	Affiche des codes de barre oblique inverse pour tous les caractères non imprimables.	Il\sexiste\squatres\types\sd'affichage.\n\\\sest\sune\sbarre\soblique\sinverse.
Affichage style mot de passe	Affiche un astérisque (*) pour chaque caractère, y compris les espaces.	***** *****
Affichage hexadécimal	Affiche la valeur ASCII de chaque caractère en hexadécimal au lieu du caractère lui-même.	496C 2065 7869 7374 6520 7175 6174 7265 2074 7970 6573 2064 2761 6666 6963 6861 6765 2E0A 5C20 6573 7420 756E 6520 6261 7272 6520 6F62 6C69 7175 6520 696E 7665 7273 652E

Tables

Utilisez la commande table, sur la palette **Commandes»Liste & Table**, pour créer une table sur la face-avant. Chaque cellule d'une table est une chaîne et chaque cellule se trouve dans une colonne et une ligne. Ainsi, une table correspond à l'affichage d'un tableau 2D de chaînes. La figure 9-1 présente une table avec toutes ses composantes. Reportez-vous à la section [Tableaux](#) de ce chapitre, pour plus d'informations sur les tableaux.

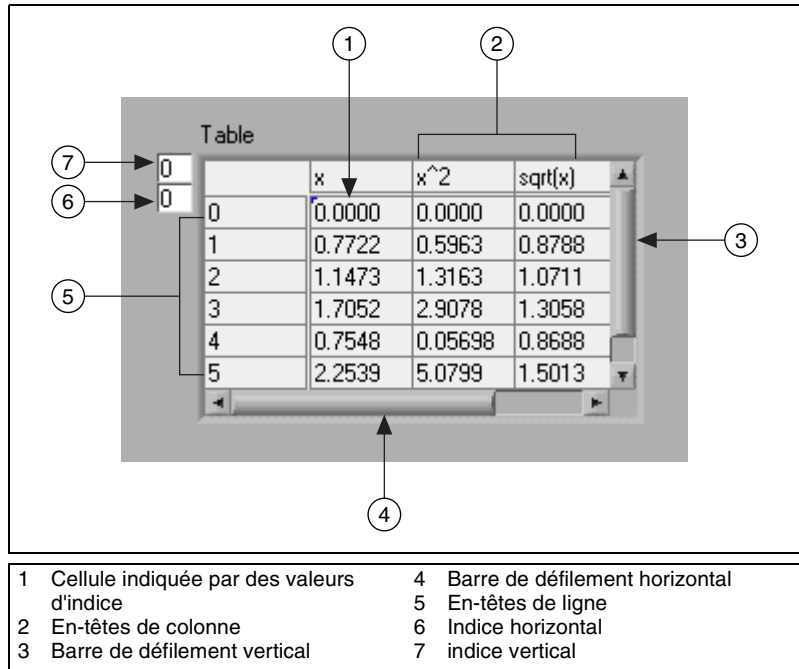


Figure 9-1. Composantes d'une table

Édition de chaînes par programme

Utilisez les fonctions Chaîne, qui se trouvent dans la palette **Fonctions»Chaîne**, pour éditer des chaînes de caractères comme suit :

- Rechercher, récupérer ou remplacer des caractères ou des sous-chaînes dans une chaîne.
- Mettre le texte entier d'une chaîne en majuscules ou minuscules.
- Rechercher et récupérer des modèles correspondants dans une chaîne.
- Récupérer une ligne d'une chaîne.
- Faire pivoter et inverser du texte dans une chaîne.

- Concaténer plusieurs chaînes.
- Supprimer des caractères dans une chaîne.

Reportez-vous à la bibliothèque `examples\general\strings.llb` pour parcourir des exemples d'utilisation des fonctions Chaîne servant à éditer des chaînes.

Formatage de chaînes

Pour utiliser des données dans d'autres VIs, fonctions ou applications, vous devez souvent convertir les données en une chaîne, puis formater la chaîne pour qu'elle soit lisible par ces VIs, fonctions ou applications. Par exemple, Microsoft Excel n'accepte que des chaînes qui comprenant des délimiteurs pour séparer les nombres ou les mots dans des cellules.

Par exemple, pour écrire un tableau 1D de valeurs numériques sur un tableur en utilisant la fonction Écrire dans un fichier, vous devez formater le tableau dans une chaîne et séparer chaque valeur numérique avec un délimiteur, comme une tabulation. Pour écrire un tableau de valeurs numériques sur un tableur en utilisant le VI Écrire dans un fichier tableur, vous devez formater le tableau avec la fonction Tableau en chaîne au format tableur et spécifier un format et un délimiteur.

Utilisez les fonctions Chaîne, qui se trouvent dans la palette

Fonctions»Chaîne pour accomplir les tâches suivantes :

- Concaténer plusieurs chaînes.
- Extraire un sous-ensemble de chaînes d'une chaîne.
- Convertir des données en chaînes.
- Formater une chaîne pour l'utiliser avec un traitement de texte ou un tableur.

Utilisez les VIs et les fonctions d'E/S sur fichiers, situés dans la palette **Fonctions»E/S sur fichiers**, pour enregistrer les chaînes dans des fichiers texte ou tableur.

Spécificateurs de format

Dans de nombreux cas, vous devez entrer un ou plusieurs spécificateurs de format dans le paramètre **chaîne de format** d'une fonction Chaîne pour formater une chaîne. Un code de spécificateur de format indique comment convertir des données en chaînes et vice versa. LabVIEW utilise les codes

de conversion pour déterminer le format textuel du paramètre. Par exemple, un spécificateur de format %x convertit un entier hexadécimal en chaîne et inversement.

Les fonctions Formater en chaîne et Balayer une chaîne peuvent utiliser plusieurs spécificateurs de format dans le paramètre **chaîne de format**, un pour chaque entrée ou sortie de la fonction redimensionnable.

Les fonctions Tableau en chaîne au format tableur et Chaîne au format tableur en tableau n'utilisent qu'un spécificateur de format dans le paramètre **chaîne de format** parce que ces fonctions n'ont qu'une entrée à convertir. LabVIEW traite tout spécificateur supplémentaire que vous insérez dans ces fonctions comme des chaînes littérales, sans signification particulière.

Numériques et chaînes

Les données numériques et les données chaîne diffèrent parce que les données chaîne correspondent à des caractères ASCII, et pas les données numériques. Les fichiers texte et tableur acceptent uniquement des chaînes. Pour écrire des données numériques dans un fichier texte ou tableur, vous devez d'abord convertir les données numériques en une chaîne.

Pour ajouter un jeu de valeurs numériques à une chaîne, convertissez les données numériques en une chaîne et utilisez la fonction Concaténer des chaînes ou une autre fonction Chaîne pour ajouter de nouvelles chaînes à la chaîne existante. Utilisez les fonctions Conversion chaîne/nombre, situées dans la palette **Fonctions»Chaîne»Conversions chaîne/nombre**, pour convertir des valeurs numériques en chaînes.

Une chaîne peut inclure un jeu de valeurs numériques que vous affichez dans un graphe ou un diagramme. Par exemple, vous pouvez lire un fichier texte qui inclut un jeu de valeurs numériques que vous souhaitez positionner sur un diagramme. Cependant, comme ces valeurs numériques sont en texte ASCII, vous devez lire les valeurs numériques comme une chaîne, puis formater la chaîne en un jeu de valeurs numériques avant de positionner les valeurs numériques sur un diagramme.

La figure 9-2 présente une chaîne qui inclut un jeu de valeurs numériques, convertit la chaîne en valeurs numériques, construit un tableau de valeurs numériques et positionne les valeurs numériques sur un graphe.

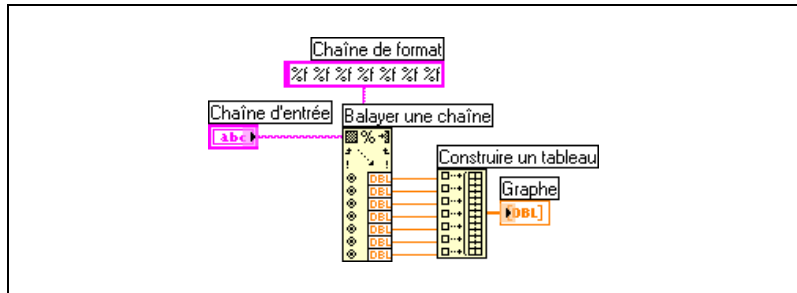


Figure 9-2. Conversion d'une chaîne en valeurs numériques

Conversion de types de données en XML

Extensible Markup Language (XML) est une norme de formatage qui utilise des tags pour décrire les données. Contrairement à un tag HTML, un tag XML n'indique pas au navigateur comment formater les données. Au lieu de cela, le tag XML identifie les données.

Par exemple, imaginez que vous êtes un libraire qui vend des livres sur le Web. Vous voulez classer chaque livre de votre librairie selon les critères suivants :

- Type de livre
- Titre
- Auteur
- Éditeur
- Prix
- Genre(s)
- Résumé
- Nombre de pages

Vous pouvez créer un fichier XML pour chaque livre. Le fichier XML d'un livre intitulé *Visiter les belles cathédrales de France* serait similaire à ceci :

```
<documentaire>
<Titre>Visiter les belles cathédrales de France</Titre>
<Auteur>Tony Walters</Auteur>
<Éditeur>Douglas Drive Publishing</Éditeur>
<PrixEuros>29.99</PrixEuros>
<Genre>Voyage</Genre>
<Genre>Architecture</Genre>
<Genre>Histoire</Genre>
<Résumé>Ce livre illustre de manière détaillée douze des
plus belles cathédrales françaises avec des
photographies en couleur, des coupes transversales à
l'échelle et l'historique de leur construction.</Résumé>
<Pages>224</Pages>
</documentaire>
```

De manière similaire, vous pouvez classer des données LabVIEW par nom, valeur et type. Vous pouvez représenter en format XML une commande chaîne pour un nom d'utilisateur de la façon suivante :

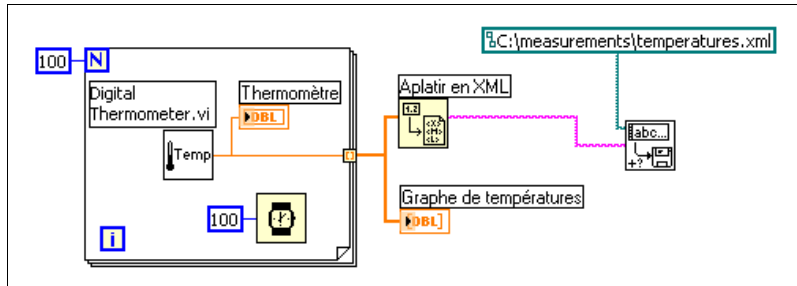
```
<Chaîne>
<Nom>Nom de l'utilisateur</Nom>
<Valeur>Reggie Harmon</Valeur>
</Chaîne>
```

Utilisation de types de données basés sur le format XML

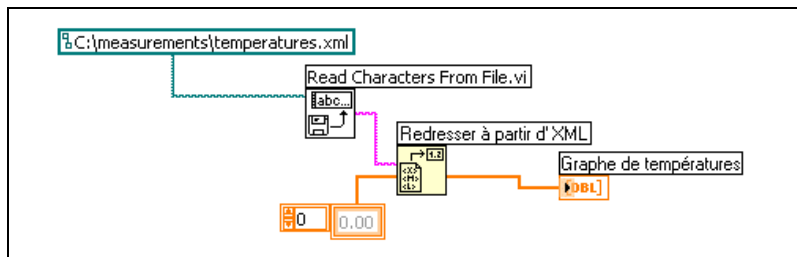
La conversion de données LabVIEW en XML formate les données de telle manière que lorsque vous les enregistrez dans un fichier, vous pouvez aisément identifier leur valeur, nom et type à partir des tags les décrivant. Par exemple, si vous convertissez un tableau de valeurs de température en format XML et que vous enregistrez ces données dans un fichier texte, vous pouvez facilement identifier les valeurs de température en repérant le tag `<Valeur>` qui identifie chaque température.

Utilisez la fonction **Aplatir en XML** située sur la palette **Fonctions» Avancé» Manipulation de données** pour convertir un type de donnée LabVIEW en format XML. L'exemple suivant génère 100 températures

simulées, trace les tableaux de températures dans un graphe, convertit le tableau de nombres en format XML et écrit les données XML dans le fichier `temperatures.xml`.



Utilisez la fonction Redresser à partir d'XML, située dans la palette **Fonctions»Avancé»Manipulation de données**, pour convertir des données de format XML en données de type LabVIEW. L'exemple suivant lit les 100 températures contenues dans le fichier `temperatures.xml` et trace le tableau de températures sur une courbe.



Remarque Bien que vous puissiez aplatir des données LabVIEW de type variant en format XML, essayer de redresser des données variant à partir du format XML donne lieu à un variant LabVIEW vide.

Schéma XML de LabVIEW

LabVIEW convertit des données selon un schéma XML établi. Actuellement, vous ne pouvez pas créer des schémas personnalisés, ni contrôler la façon dont LabVIEW assigne des tags aux données. Par ailleurs, vous ne pouvez pas convertir la totalité de VIs ou de fonctions en XML.

Groupement des données avec des tableaux et des clusters

Utilisez les commandes et fonctions de tableaux et clusters situées dans les palettes **Commandes»Tableau et Cluster**, **Fonctions»Tableau et Fonctions»Cluster** pour grouper les données. Les tableaux regroupent les éléments de données du même type. Les clusters regroupent les éléments de données de types différents.

Tableaux

Un tableau se compose d'éléments et de dimensions. Les éléments sont les données qui constituent le tableau. La dimension correspond à la longueur, la hauteur ou la profondeur d'un tableau. Un tableau peut avoir une ou plusieurs dimensions et jusqu'à $2^{31} - 1$ éléments par dimension, en fonction de la capacité mémoire.

Vous pouvez construire des tableaux de données numériques, de booléens, de chemins, de chaînes, de waveforms et de clusters. Pensez à utiliser des tableaux lorsque vous travaillez avec beaucoup de données semblables et lorsque vous effectuez des calculs répétitifs. Les tableaux sont utiles pour stocker des données collectées dans les waveforms ou générées dans des boucles, où chaque itération d'une boucle produit un élément du tableau.

Vous ne pouvez pas créer des tableaux de tableaux. Vous pouvez néanmoins créer un tableau de clusters où chaque cluster contient un ou plusieurs tableaux. Reportez-vous à la section [Restrictions pour les tableaux](#) dans ce chapitre pour plus d'informations sur les types d'éléments supportés par les tableaux. Reportez-vous à la section [Clusters](#) pour plus d'informations sur les clusters.

Indices

Pour repérer un élément dans un tableau, vous avez besoin d'un indice par dimension. Dans LabVIEW, les indices vous permettent de naviguer dans un tableau et de récupérer des éléments, des lignes, des colonnes et des pages d'un tableau sur le diagramme.

Exemples de tableaux

Un exemple de tableau simple est un tableau de texte qui répertorie les neuf planètes de notre système solaire. LabVIEW représente ce tableau comme un tableau de chaînes 1D avec neuf éléments.

Les éléments de tableaux sont ordonnés. Un tableau utilise un indice pour que vous puissiez accéder facilement à tout élément particulier. L'indice démarre à zéro, ce qui signifie qu'il se trouve dans la gamme de 0 à $n - 1$, où n est le nombre d'éléments dans le tableau. Dans cet exemple, $n = 9$ pour les neuf planètes, et la gamme de l'indice est comprise entre 0 et 8. La terre est la troisième planète ; elle a donc l'indice 2.

Un autre exemple de tableau est un signal représenté comme un tableau numérique dans lequel chaque élément consécutif est égal à la valeur de la tension à un intervalle de temps donné, comme indiqué sur la figure 9-3.

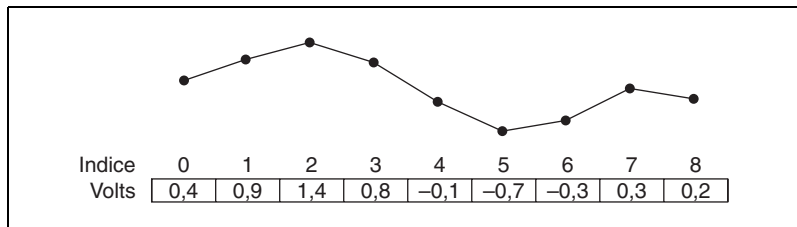


Figure 9-3. Signal dans un tableau de valeurs numériques

Un exemple plus complexe de tableau est un graphe représenté comme un tableau de points, où chaque point est un cluster contenant une paire de valeurs numériques qui représentent les coordonnées X et Y, comme indiqué sur la figure 9-4.

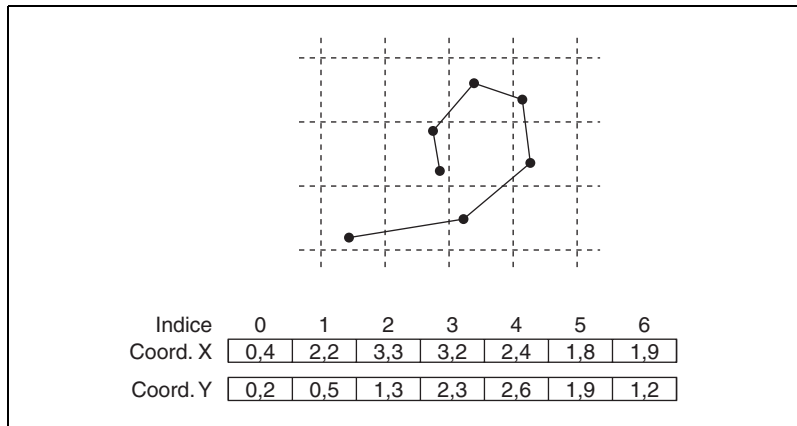


Figure 9-4. Graphe dans un tableau de points

Les exemples précédents utilisent des tableaux 1D. Un tableau 2D stocke des éléments dans une grille. Pour repérer un élément, vous avez besoin d'un indice de colonne et d'un indice de ligne, les deux commençant à zéro. La figure 9-5 présente un tableau 2D de 6 colonnes et 4 lignes, qui contient $6 \times 4 = 24$ éléments.

		Indice de colonne					
		0	1	2	3	4	5
Indice de ligne	0						
	1						
	2						
	3						

Figure 9-5. Tableau 2D de 6 colonnes et 4 lignes

Par exemple, un échiquier comporte huit colonnes et huit lignes pour un total de 64 positions. Chaque position est vide ou contient une pièce d'échec. Vous pouvez représenter un échiquier comme un tableau 2D de chaînes. Chaque chaîne correspond au nom de la pièce qui occupe la position correspondante sur l'échiquier, ou à une chaîne vide si cette position est vide.

Vous pouvez généraliser les exemples de tableaux 1D dans les figures 9-3 et 9-4 à deux dimensions en ajoutant une ligne au tableau. La figure 9-6 présente un ensemble de signaux représenté comme un tableau 2D de valeurs numériques. L'indice de ligne sélectionne le signal et l'indice de colonne sélectionne un point du signal.

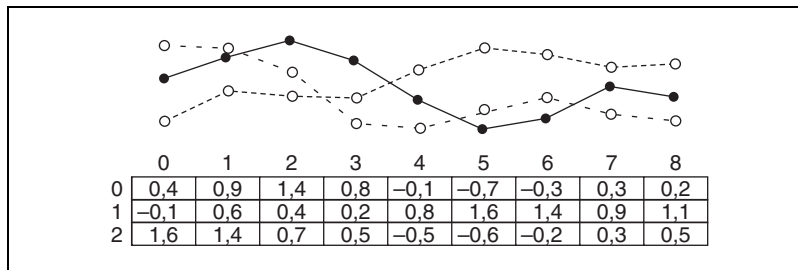


Figure 9-6. Signaux multiples dans un tableau 2D de valeurs numériques

Reportez-vous à la bibliothèque `exemples\general\arrays.llb` pour d'autres exemples de tableaux.

Restrictions pour les tableaux

Vous pouvez créer un tableau de presque n'importe quel type de données, aux exceptions près suivantes :

- Vous ne pouvez pas créer des tableaux de tableaux. Vous pouvez cependant utiliser un tableau multidimensionnel ou créer un tableau de clusters où chaque cluster contient un ou plusieurs tableaux.
- Vous ne pouvez pas créer un tableau de graphes car un graphe constitue un type de données tableau et un tableau ne peut pas contenir un autre tableau. Vous pouvez cependant avoir un tableau de graphes, autres que des graphes X-Y, s'ils se trouvent dans un cluster.
- Vous ne pouvez pas créer un tableau de graphes déroulants.

Création de commandes, d'indicateurs et de constantes tableau

Créez une commande ou un indicateur tableau sur la face-avant en y plaçant un tableau vierge, comme indiqué à la figure 9-7, et en déplaçant un élément ou un objet de données, pouvant être une commande ou un indicateur numérique, booléen, de chaîne, de chemin, de refnum ou de cluster, dans le tableau vierge.

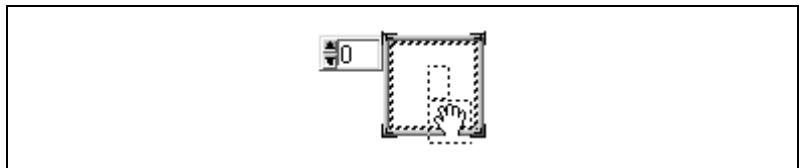


Figure 9-7. Tableau vierge

Le tableau vierge se redimensionne automatiquement pour accepter le nouvel objet, que ce soit une petite commande booléenne ou un grand graphe 3D.

Pour afficher un élément particulier sur la face-avant, tapez le numéro de l'indice dans l'afficheur de l'indice ou utilisez les flèches sur l'afficheur de l'indice pour parvenir à ce numéro.

Pour créer une constante tableau sur le diagramme, sélectionnez **Fonctions»Tableau»Constante tableau** pour placer le tableau vierge, puis une constante chaîne, une constante numérique ou une constante cluster dans le tableau vierge. Vous pouvez utiliser une constante tableau pour une comparaison avec un autre tableau.

Affichage de l'indice du tableau

Un tableau 2D contient des lignes et des colonnes. Comme présenté à la figure 9-8, l'afficheur en haut à gauche correspond à l'indice de ligne et l'afficheur en bas à gauche correspond à l'indice de colonne. L'afficheur de droite présente la valeur à la position spécifiée par les deux premiers afficheurs. La figure 9-8 indique que la valeur de la ligne 6, colonne 13, est 66.

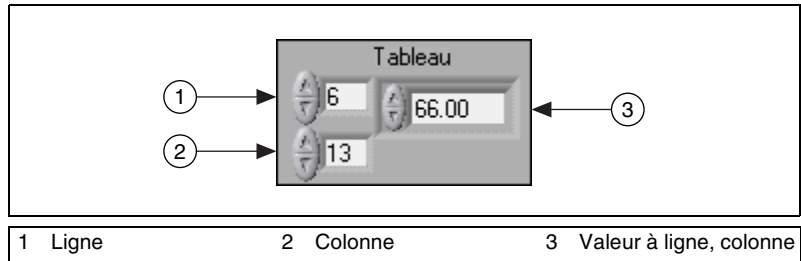


Figure 9-8. Commande tableau

Les indices de ligne et de colonne démarrent à zéro, ce qui signifie que la première colonne est la colonne 0, la deuxième colonne est la colonne 1, etc. Ainsi, lorsque vous changez les afficheurs d'indice du tableau suivant sur ligne 1 et colonne 2, une valeur de 6 s'affiche.

0	1	2	3
4	5	6	7
8	9	10	11

Si vous essayez d'afficher une colonne ou une ligne hors de la gamme des dimensions du tableau, la commande de tableau est grisée pour indiquer l'absence de valeur définie et LabVIEW affiche la valeur par défaut du type de données. La valeur par défaut du type de données dépend du type de données du tableau.

Utilisez l'outil Flèche pour afficher plusieurs lignes ou colonnes à la fois.

Fonctions de tableaux

Utilisez les fonctions Tableau, qui se trouvent dans la palette **Fonctions» Tableau** pour créer et manipuler des tableaux afin d'effectuer les tâches suivantes :

- Extraire des éléments de données d'un tableau.
- Insérer, supprimer ou remplacer des éléments de données dans un tableau.
- Diviser des tableaux.

Redimensionnement automatique des fonctions de tableaux

Les fonctions Indexer un tableau, Remplacer une portion du tableau, Insérer dans un tableau, Supprimer une portion de tableau et Sous-ensemble d'un tableau se redimensionnent automatiquement pour correspondre aux dimensions du tableau d'entrée que vous câblez. Par exemple, si vous câblez un tableau 1D à l'une de ces fonctions, la fonction affiche une entrée d'indice unique. Si vous câblez un tableau 2D à la même fonction, elle affiche deux entrées d'indice : une pour la ligne et une pour la colonne.

Vous pouvez accéder à plusieurs éléments ou à un sous-tableau (ligne, colonne ou page) avec ces fonctions, en utilisant l'outil Flèche pour redimensionner manuellement la fonction. Lorsque vous étendez une de ces fonctions, les fonctions s'étendent de façon incrémentale en fonction des dimensions du tableau câblé à la fonction. Si vous câblez un tableau 1D à l'une de ces fonctions, la fonction s'agrandit d'une entrée d'indice unique. Si vous câblez un tableau 2D à la même fonction, elle s'agrandit de deux entrées d'indice : une pour la ligne et une pour la colonne.

Les entrées d'indice que vous câblez déterminent la forme du sous-tableau que vous voulez atteindre ou modifier. Par exemple, si l'entrée sur une fonction Indexer un tableau est un tableau 2D et que vous ne câblez que l'entrée **indice (ligne)**, une ligne 1D entière est extraite du tableau. Si vous ne câblez que l'entrée **indice (col)**, une colonne 1D entière est extraite du tableau. Si vous câblez l'entrée **indice (ligne)** et l'entrée **indice (col)**, un élément unique est extrait du tableau. Chaque groupe d'entrées est indépendant et peut accéder à n'importe quelle partie d'une dimension du tableau.

Le diagramme présenté en figure 9-9 utilise la fonction Indexer un tableau pour récupérer une ligne et un élément d'un tableau 2D.

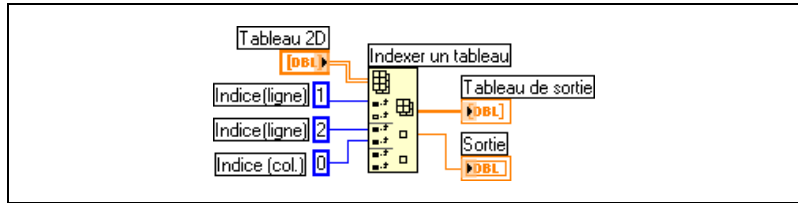


Figure 9-9. Indexation d'un tableau 2D

Pour accéder à plusieurs valeurs consécutives dans un tableau, étendez la fonction Indexer un tableau, mais ne câblez pas les valeurs aux entrées de l'indice dans chaque incrément. Par exemple, pour récupérer les première, deuxième et troisième lignes d'un tableau 2D, étendez la fonction Indexer un tableau de trois incréments et câblez les indicateurs de tableau 1D à chaque sortie du **sous-tableau**.

Clusters

Les clusters regroupent des éléments de données de types différents, comme un assemblage de fils de liaison dans un câble de téléphone, où chaque fil de liaison dans le câble représente un élément différent du cluster. Un cluster est similaire à une structure dans les langages de programmation textuels.

L'assemblage de plusieurs éléments de données dans des clusters élimine l'encombrement des câbles sur le diagramme et réduit le nombre de terminaux de connexion nécessaires pour les sous-VIs. Le connecteur possède 28 terminaux maximum. Si votre face-avant contient plus de 28 commandes et indicateurs que vous souhaitez utiliser, groupez certains d'entre eux dans un cluster et affectez le cluster à un terminal sur le connecteur.

Bien que les éléments de cluster et de tableau soient ordonnés, vous devez désassembler tous les éléments du cluster à la fois. Vous pouvez également utiliser la fonction Désassembler par nom pour accéder à des éléments spécifiques d'un cluster. Les clusters diffèrent également des tableaux dans la mesure où leur taille est fixe. Comme un tableau, un cluster est soit une commande soit un indicateur. Un cluster ne peut pas contenir simultanément des commandes et des indicateurs.

La plupart des clusters sur le diagramme possèdent un modèle de fil de liaison et une icône de type de données roses. Les clusters de valeurs

numériques, appelés parfois des points, possèdent un modèle de fil de liaison et une icône de type de données marron. Vous pouvez câbler les clusters numériques marron aux fonctions numériques, comme Ajouter ou Racine carrée, pour réaliser la même opération simultanément sur tous les éléments du cluster.

Les éléments de cluster possèdent un ordre logique qui n'est pas lié à leur position dans le cluster. Le premier objet que vous placez dans le cluster est l'élément 0, le deuxième est l'élément 1 et ainsi de suite. Si vous supprimez un élément, l'ordre est automatiquement ajusté. L'ordre du cluster détermine l'ordre dans lequel les éléments apparaissent aux terminaux des fonctions Assembler et Désassembler sur le diagramme. Vous pouvez visualiser et modifier l'ordre du cluster en effectuant un clic droit sur le cadre du cluster et en sélectionnant **Ordonner les commandes dans le cluster...** dans le menu local.

Les clusters doivent avoir le même nombre d'éléments pour pouvoir être câblés. Les éléments correspondants, déterminés par l'ordre du cluster, doivent avoir des types de données compatibles. Par exemple, si un nombre flottant double précision contenu dans un cluster correspond, d'un point de vue ordre de cluster, à une chaîne contenue dans un autre cluster, le fil de liaison sur le diagramme apparaît brisé et le VI ne s'exécute pas. Si vous utilisez des valeurs numériques ayant des représentations différentes dans le même cluster, LabVIEW les contraint à la même représentation. Reportez-vous à la section [Conversion numérique](#) de l'annexe B, [Fonctions polymorphes](#), pour plus d'informations sur la conversion numérique.

Utilisez les fonctions Cluster, qui se trouvent dans la palette **Fonctions» Cluster** pour créer et manipuler des clusters afin d'effectuer les tâches suivantes :

- Extraire des éléments de données individuels d'un cluster.
- Ajouter des éléments de données individuels à un cluster.
- Réduire un cluster à ses éléments de données individuels.

Variables globales et locales

Dans LabVIEW, vous lisez ou écrivez des données dans un objet de la face-avant en utilisant son terminal de diagramme. Un objet de face-avant n'a qu'un terminal de diagramme et votre application peut avoir besoin d'accéder aux données dans ce terminal depuis plusieurs emplacements.

Les variables globales et locales transmettent des informations entre des emplacements dans votre application qui ne peuvent pas être connectés par un fil de liaison. Utilisez des variables locales pour accéder à des objets de la face-avant à partir de plusieurs emplacements dans un même VI. Utilisez des variables globales pour accéder à des données et les transmettre parmi plusieurs VIs.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour plus d'informations sur l'utilisation des variables locales et globales.

Variables locales

Utilisez les variables locales pour accéder à des objets de la face-avant à partir de plusieurs emplacements dans un même VI et transmettre les données entre des structures du diagramme que vous ne pouvez pas connecter avec un fil de liaison.

Avec une variable locale, vous pouvez lire ou écrire dans une commande ou un indicateur sur la face-avant. Écrire dans une variable locale revient à transmettre des données à un autre terminal. Cependant, avec une variable locale, vous pouvez écrire même s'il s'agit d'une commande ou lire même s'il s'agit d'un indicateur. En effet, avec une variable locale, vous pouvez accéder à des objets de la face-avant à la fois comme entrée et comme sortie.

Par exemple, si l'interface utilisateur exige la connexion des utilisateurs, vous pouvez effacer les invites **Login** et **Mot de passe** à chaque fois qu'un nouvel utilisateur se connecte. Utilisez une variable locale pour lire les commandes de chaînes **Login** et **Mot de passe** lorsqu'un utilisateur se

connecte et pour écrire des chaînes vides dans ces commandes lorsque l'utilisateur se déconnecte.

Création de variables locales

Cliquez avec le bouton droit sur un objet de la face-avant ou un terminal du diagramme et sélectionnez **Créer»Variable locale** dans le menu local pour créer une variable locale. Une icône pour la variable locale associée à l'objet apparaît sur le diagramme.



Vous pouvez aussi sélectionner **Fonctions»Structures»Variable locale** et placer une variable locale sur le diagramme. Le nœud de variable locale, présenté sur la gauche, n'est pas encore associé à une commande ou à un indicateur. Cliquez avec le bouton droit sur le nœud de variable locale et sélectionnez **Sélectionner un élément** dans le menu local pour indiquer quel objet de la face-avant vous désirez associer à la variable locale. Le sous-menu local **Sélectionner un élément** répertorie tous les objets de la face-avant du VI global qui possèdent des étiquettes.

LabVIEW utilise ces étiquettes pour associer des variables locales à des objets de la face-avant ; veuillez donc étiqueter les commandes et indicateurs de la face-avant avec ce type d'étiquettes descriptives.

Reportez-vous à la section [Étiquetage](#) du chapitre 4, [Construction de la face-avant](#), pour plus d'informations sur les étiquettes libres et celles appartenant à un objet.

Variables globales

Utilisez des variables globales pour accéder à des données et les transmettre à plusieurs VIs qui s'exécutent simultanément. Les variables globales sont des objets intégrés à LabVIEW. Lorsque vous créez une variable globale, LabVIEW crée automatiquement un VI global spécial, qui possède une face-avant mais pas de diagramme. Ajoutez des commandes et indicateurs à la face-avant du VI global pour définir les types de données des variables globales qu'il contient. En effet, cette face-avant est un conteneur depuis lequel plusieurs VIs peuvent accéder aux données.

Par exemple, supposons que deux VIs s'exécutent simultanément. Chaque VI contient une boucle While et écrit des points de données sur un graphe déroulant. Le premier VI contient une commande booléenne pour terminer les deux VIs. Vous devez utiliser une variable globale pour terminer les deux boucles avec une commande booléenne unique. Si les deux boucles se trouvent sur un diagramme unique dans le même VI, vous pouvez utiliser une variable locale pour terminer les boucles.

Création de variables globales



Sélectionnez **Fonctions»Structures»Variable globale** et placez une variable globale, ci-contre, sur le diagramme. Double-cliquez sur le nœud de variable globale pour afficher la face-avant du VI global. Mettez des commandes et des indicateurs sur cette face-avant comme vous le feriez pour une face-avant standard.

LabVIEW utilise les étiquettes pour identifier des variables globales ; c'est pourquoi vous devez étiqueter les commandes et indicateurs de la face-avant avec des étiquettes descriptives. Reportez-vous à la section [Étiquetage](#) du chapitre 4, [Construction de la face-avant](#), pour plus d'informations sur les étiquettes libres et celles appartenant à un objet.

Vous pouvez créer plusieurs VIs globaux uniques, chacun avec un objet de la face-avant, ou vous pouvez créer un VI global avec plusieurs objets en face-avant. Un VI global avec plusieurs objets est plus efficace parce que vous pouvez grouper des variables apparentées entre elles. Le diagramme d'un VI peut inclure plusieurs nœuds de variable globale associés aux commandes et indicateurs sur la face-avant d'un VI global. Ces nœuds de variable globale sont soit des copies du premier nœud de variable globale placé sur le diagramme du VI global, soit des nœuds de variable globale des VIs globaux placés sur le VI actuel. Vous pouvez placer des VIs globaux sur d'autres VIs de la même façon que vous placez des sous-VIs sur d'autres VIs. Chaque fois que vous mettez un nouveau nœud de variable globale sur un diagramme, LabVIEW crée un nouveau VI associé uniquement à ce nœud de variable globale et à ses copies. Reportez-vous à la section [Sous-VIs](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour plus d'informations sur les sous-VIs.

Après avoir placé tous les objets sur la face-avant du VI global, enregistrez-le et retournez au diagramme du VI original. Vous devez ensuite sélectionner l'objet dans le VI global auquel vous voulez accéder. Cliquez avec le bouton droit sur le nœud de variable globale et sélectionnez un objet de la face-avant dans le menu local **Sélectionner un élément**. Le sous-menu local **Sélectionner un élément** répertorie tous les objets de la face-avant du VI global ayant des étiquettes liées.

VARIABLES DE LECTURE ET D'ÉCRITURE

Après avoir créé une variable locale ou globale, vous pouvez lire des données d'une variable ou écrire des données dans une variable. Par défaut, une nouvelle variable reçoit des données. Ce genre de variable fonctionne comme un indicateur et correspond à une variable globale ou locale

d'écriture. Lorsque vous écrivez de nouvelles données sur une variable globale ou locale, la commande ou l'indicateur de la face-avant associé se met à jour avec les nouvelles données.

Vous pouvez également configurer une variable pour qu'elle se comporte comme une source de données ou une variable de lecture locale ou globale. Cliquez avec le bouton droit sur la variable et sélectionnez **Changer en lecture** dans le menu local pour configurer la variable comme une commande. Lorsque ce nœud s'exécute, le VI lit les données dans la commande ou l'indicateur associé de la face-avant.

Pour changer la variable pour qu'elle reçoive des données du diagramme plutôt que d'en fournir, cliquez avec le bouton droit sur la variable et sélectionnez **Changer en écriture** dans le menu local.

Sur le diagramme, vous pouvez différencier les variables locales ou globales de lecture des variables locales ou globales d'écriture, tout comme vous distinguez les commandes des indicateurs. Une variable locale ou globale de lecture possède une bordure épaisse similaire à celle d'une commande. Une variable locale ou globale d'écriture possède une bordure fine similaire à celle d'un indicateur.

Reportez-vous aux bibliothèques `exemples\general\locals.llb` et `exemples\general\globals.llb` pour parcourir des exemples d'utilisation de variables globales et locales.

Utilisation mesurée des variables globales et locales

Les variables globales et locales constituent des concepts avancés de LabVIEW. Elles ne font pas partie de manière inhérente du modèle d'exécution en flux de données LabVIEW. Les diagrammes peuvent devenir difficiles à lire lorsque vous utilisez des variables globales et locales ; c'est pourquoi vous devez les utiliser avec précaution. Une mauvaise utilisation des variables globales et locales, comme les utiliser à la place d'un connecteur ou pour accéder à des valeurs dans chaque cadre d'une structure Séquence, peut entraîner un comportement indéterminé des VIs. Une utilisation intensive des variables globales et locales, comme les utiliser à la place du câblage ou du flux de données, ralentit l'exécution. Reportez-vous à la section [Flux des données dans le diagramme](#) du chapitre 5, [Construction du diagramme](#), pour plus d'informations sur le modèle d'exécution par flux de données sous LabVIEW.

Initialisation des variables globales et locales

Vérifiez que les variables globales et locales contiennent des données connues avant de démarrer l'exécution du VI. Sinon, les variables peuvent contenir des données qui génèrent un comportement incorrect du VI.

Si vous n'initialisez pas la variable avant que le VI la lise pour la première fois, la variable contient la valeur par défaut de l'objet de face-avant associé.

Situations de compétition

Comme les VIs suivent un modèle d'exécution par flux de données, les variables globales et locales de LabVIEW ne se comportent pas comme les variables globales et locales des langages de programmation textuels. Une situation de compétition se produit lorsque deux parties du code ou davantage, qui s'exécutent en parallèle, changent la valeur de la même ressource partagée, généralement une variable globale ou locale. La figure 10-1 présente un exemple de situation de compétition.

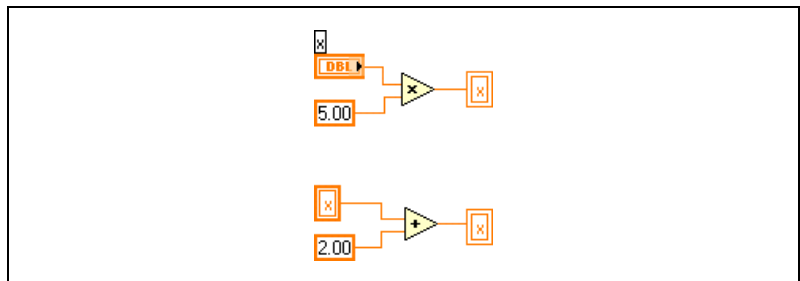


Figure 10-1. Situations de compétition

La sortie de ce VI dépend de l'ordre d'exécution des opérations. Comme il n'existe pas de dépendance de données entre les deux opérations, il est impossible de déterminer laquelle s'exécute en premier lieu. Pour éviter des situations de compétition, n'écrivez pas sur une variable que vous lisez. Reportez-vous à la section *Dépendance des données et dépendance des données artificielle* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur la dépendance des données.

Si vous utilisez des variables globales dans des VIs qui s'exécutent en parallèle, vous pouvez utiliser une variable globale booléenne pour indiquer quand les données des variables globales changent. D'autres VIs peuvent surveiller cette variable globale booléenne pour déterminer si les données ont changé et lire la nouvelle valeur.

Considérations sur la mémoire lors de l'utilisation de variables locales

Lorsque vous créez des sous-VIs, vous créez un connecteur qui décrit comment transmettre des données avec le sous-VI. Le connecteur n'effectue pas de copie des buffers de données des VIs appelants.

Les variables locales effectuent des copies des buffers de données. Lorsque vous lisez une variable locale, vous créez un nouveau buffer pour les données depuis la commande associée.

Si vous utilisez des variables locales pour transférer de grandes quantités de données d'un endroit du diagramme à un autre, vous utilisez généralement plus de mémoire, et de ce fait, la vitesse d'exécution est plus lente que si vous transfériez des données avec un fil de liaison.

Considérations sur la mémoire lors de l'utilisation de variables globales

Lorsque vous lisez depuis une variable globale, LabVIEW crée une copie des données stockées dans cette variable globale.

Lorsque vous manipulez de grands tableaux et chaînes, la durée et la mémoire nécessaires pour manipuler les variables globales peuvent être considérables. Manipuler des variables globales est particulièrement inefficace lorsque vous utilisez des tableaux, parce que si vous ne modifiez qu'un seul élément de tableau, LabVIEW stocke le tableau entier. Si vous lisez la variable globale à plusieurs endroits dans l'application, vous créez plusieurs buffers, ce qui est inefficace et ralentit l'exécution.

Graphes et graphes déroulants

Utilisez les graphes et les graphes déroulants pour afficher des tracés de données sous forme de graphiques.

Les graphes et les graphes déroulants ont une façon différente d'afficher et de mettre à jour les données. Les VIs utilisant des graphes collectent généralement les données dans un tableau puis les tracent sur le graphe, comme un tableur stocke d'abord les données avant d'en générer le tracé. À l'inverse, un graphe déroulant ajoute de nouveaux points de données à l'affichage. Sur un graphe déroulant, vous pouvez afficher la mesure actuelle dans son contexte, avec les données déjà acquises.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour plus d'informations sur l'utilisation des graphes et graphes déroulants.

Types de graphes et de graphes déroulants

Les graphes et les graphes déroulants, situés dans la palette **Commandes» Graphe**, comprennent les types suivants :

- **Graphe déroulant et graphe** : affichent les données acquises à une vitesse constante
- **Graphe XY** : affiche les données acquises à une vitesse non constante, comme des données acquises sur déclenchement.
- **Graphe déroulant d'intensité et graphe d'intensité** : affichent les données 3D sur un tracé 2D en utilisant des couleurs pour afficher les valeurs de la troisième dimension.
- **Graphe numérique**: affiche les données comme des impulsions ou des groupes de lignes numériques. Les ordinateurs transfèrent les données numériques par impulsions à d'autres ordinateurs.
- **(Windows) Graphes 3D** : affichent des données 3D sur un tracé 3D dans un objet ActiveX de la face-avant.

Reportez-vous à la bibliothèque `examples\general\graphs.lib` pour des exemples de graphes et de graphes déroulants.

Options de graphe et de graphe déroulant

Bien que les graphes et les graphes déroulants positionnent les données différemment, ils ont en commun plusieurs options auxquelles vous accédez depuis le menu local. Reportez-vous aux sections [Personnalisation des graphes](#) et [Personnalisation des graphes déroulants](#) dans ce chapitre, pour plus d'informations sur les options spécifiques aux graphes ou aux graphes déroulants.

Les graphes XY, les graphes et les graphes déroulants possèdent des options différentes de celles des graphes et graphes déroulants 3D, numériques et d'intensité. Reportez-vous aux sections [Graphes et graphes déroulants d'intensité](#), [Graphes 3D](#) et [Graphes numériques](#) dans ce chapitre, pour plus d'informations sur les options spécifiques aux graphes et graphes déroulants 3D, numériques et d'intensité.

Échelles X et Y multiples sur des graphes et des graphes déroulants

Tous les graphes et graphes déroulants supportent plusieurs échelles x et y. Utilisez plusieurs échelles sur un graphe ou un graphe déroulant pour afficher plusieurs courbes qui ne partagent pas une échelle x ou y commune. Cliquez avec le bouton droit sur l'échelle x ou y du graphe ou du graphe déroulant, puis sélectionnez **Dupliquer l'échelle** dans le menu local pour ajouter plusieurs échelles x ou y sur le graphe ou le graphe déroulant.

Lissage pour les graphes et graphes déroulants

Vous pouvez améliorer l'apparence des courbes des graphes déroulants et des graphes en utilisant le lissage. Lorsque vous activez le lissage, les courbes apparaissent plus lisses. Le lissage n'altère pas la largeur de ligne, le style de ligne ou le style de point.



Remarque Le lissage n'est pas disponible pour les graphes numériques.

Pour activer le lissage, cliquez avec le bouton droit sur la légende de la courbe et sélectionnez **Lissage** dans le menu local. Si la légende de la courbe n'est pas visible, cliquez avec le bouton droit sur le graphe déroulant ou le graphe et sélectionnez **Éléments visibles** » **Légende de la courbe** dans le menu local.



Remarque Comme le lissage utilise beaucoup de calculs, il risque de ralentir les performances.

Personnalisation de l'apparence des graphes et des graphes déroulants

Personnalisez l'apparence des graphes et des graphes déroulants en affichant ou en masquant des options. Cliquez avec le bouton droit sur le graphe déroulant ou le graphe et sélectionnez **Éléments visibles** dans le menu local pour afficher ou masquer les options suivantes :

- **Légende de la courbe** : définit la couleur et le style des courbes. Redimensionne la légende pour afficher plusieurs courbes.
- **Légende de l'échelle** : définit les étiquettes des échelles et configure les propriétés des échelles.
- **Palette du graphe** : change l'échelle et le format pendant l'exécution d'un VI.
- **Échelle X et Échelle Y** : formate les échelles x et y. Reportez-vous à la section *Options d'échelle* dans ce chapitre, pour plus d'informations sur les échelles.
- **Légende du curseur (graphe uniquement)** : affiche un repère à une coordonnée de point définie. Vous pouvez afficher plusieurs curseurs sur un graphe.
- **Barre de défilement (graphe déroulant uniquement)** : fait défiler les données dans le graphe déroulant. Utilisez la barre de défilement pour visualiser des données qui ne sont pas affichées par le buffer.

Personnalisation des graphes

Vous pouvez modifier le comportement des curseurs du graphe, les options d'échelle et les axes du graphe. La figure 11-1 présente les éléments d'un graphe.

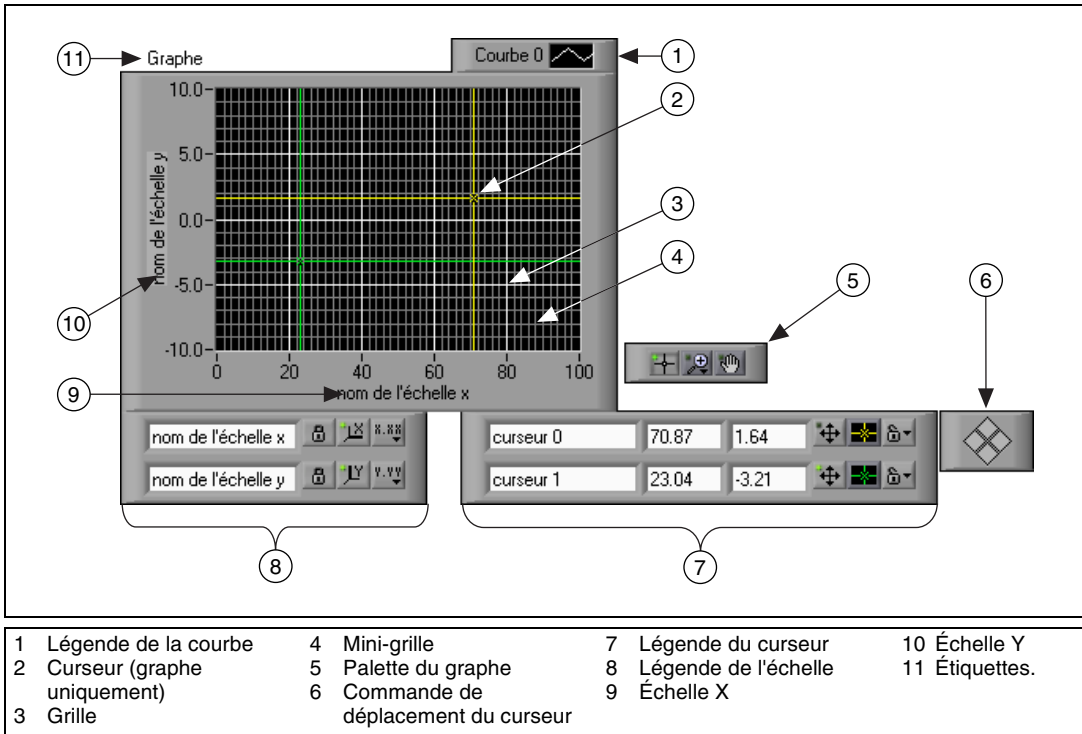


Figure 11-1. Éléments d'un graphe

Vous ajoutez la plupart des éléments listés dans la légende ci-dessus en cliquant avec le bouton droit sur le graphe, en sélectionnant **Éléments visibles** dans le menu local, puis l'élément approprié.

Curseurs de graphe

Les curseurs sur des graphes vous permettent de lire la valeur exacte d'un point sur une courbe. La valeur du curseur s'affiche dans la légende du curseur. Ajoutez un curseur à un graphe en cliquant avec le bouton droit sur le graphe, en sélectionnant **Éléments visibles»Légende du curseur** dans le menu local, puis en cliquant n'importe où dans une ligne de la légende du curseur pour activer un curseur. Utilisez l'outil Flèche pour étendre la légende du curseur à plusieurs curseurs.

Vous pouvez placer des curseurs et un affichage de curseur sur tous les graphes et vous pouvez étiqueter le curseur sur la courbe. Vous pouvez verrouiller un curseur et vous pouvez déplacer plusieurs curseurs à la fois. Un graphe peut avoir n'importe quel nombre de curseurs.

Reportez-vous au VI Zoom Graph dans la bibliothèque `exemples\general\graphs\zoom.llb` pour visualiser un exemple de lecture des valeurs de curseur et pour faire un zoom avant et arrière d'un graphe par programmation en utilisant les curseurs.

Options d'échelle

Les graphes peuvent automatiquement régler leurs échelles horizontales et verticales pour refléter les données. Ce comportement s'appelle la mise à l'échelle automatique. Activez ou désactivez la mise à l'échelle automatique en cliquant avec le bouton droit sur le graphe et en sélectionnant **Échelle X»Mise à l'échelle automatique des X** ou **Échelle Y»Mise à l'échelle automatique des Y** dans le menu local. Par défaut, la mise à l'échelle automatique est activée pour les graphes. Cependant, la mise à l'échelle automatique risque de ralentir les performances.

Utilisez l'outil Doigt ou Texte pour changer directement l'échelle horizontale ou verticale.

Légende de l'échelle d'un graphe

Utilisez la légende de l'échelle pour étiqueter les échelles et configurer leurs propriétés.



Utilisez l'outil Doigt pour cliquer sur le bouton **Format de l'échelle**, présenté sur la gauche, pour configurer le format, la précision et le mode de représentation des échelles.



Utilisez le bouton **Verrouillage de l'échelle**, présenté sur la gauche, pour activer ou désactiver la mise à l'échelle automatique pour chaque échelle, la visibilité des échelles, les étiquettes des échelles et les courbes, de même que pour formater les étiquettes des échelles, les grilles, les lignes et les couleurs des grilles.

Formatage des échelles d'un graphe

Vous pouvez formater les échelles d'un graphe pour représenter une amplitude ou un temps relatif ou absolu. Utilisez le format de temps absolu pour afficher l'heure et/ou la date sur une échelle. Si vous ne souhaitez pas

que LabVIEW définisse une date, utilisez le format de temps relatif. Pour sélectionner le format de temps relatif ou absolu, cliquez avec le bouton droit sur le graphe, sélectionnez l'échelle que vous désirez modifier, puis sélectionnez **Formatage...** dans le menu local. La boîte de dialogue **Formatage**, dans laquelle vous pouvez spécifier différentes propriétés de l'échelle du graphe, apparaît. Par défaut, l'axe des X représente le temps relatif et l'axe des Y représente l'amplitude.

Utilisation du rafraîchissement progressif

Cliquez avec le bouton droit sur le graphe et sélectionnez **Avancé»Rafraîchissement progressif** dans le menu local, pour utiliser un buffer hors écran afin de minimiser le scintillement. L'utilisation du **rafraîchissement progressif** peut dégrader les performances, en fonction de l'ordinateur et du système vidéo utilisés.

Personnalisation des graphes déroulants

À l'inverse des graphes qui affichent une waveform entière écrasant les données déjà stockées, les graphes déroulants se mettent à jour périodiquement et conservent l'historique des données préalablement stockées.

Vous pouvez personnaliser les graphes déroulants et les graphes déroulants d'intensité sur la palette **Commandes»Graphe** pour les faire correspondre à vos besoins d'affichage des données ou pour afficher plus d'informations. Parmi les options disponibles pour les graphes déroulants, figurent une barre de défilement, une légende, une palette, un afficheur numérique et la représentation des échelles selon le temps. Vous pouvez modifier le comportement de la longueur de l'historique, des modes de rafraîchissement et de l'affichage des courbes.

Longueur de l'historique

LabVIEW stocke les points de données déjà ajoutés au graphe déroulant dans un buffer ou dans l'historique du graphe déroulant. La taille par défaut d'un buffer d'historique est 1 024 points de données. Vous pouvez configurer le buffer d'historique en cliquant avec le bouton droit sur le graphe déroulant et en sélectionnant **Longueur de l'historique** dans le menu local. Vous pouvez afficher des données collectées précédemment en utilisant la barre de défilement du graphe déroulant. Cliquez avec le bouton droit sur le graphe déroulant et sélectionnez **Éléments visibles»Barre de défilement** dans le menu local pour afficher une barre de défilement.

Mode de mise à jour du graphe déroulant

Les graphes déroulants utilisent trois modes de défilement des données différents. Cliquez avec le bouton droit sur le graphe déroulant et sélectionnez **Avancé»Mode de MaJ** (Mise à jour) dans le menu local. Sélectionnez **Graphe déroulant**, **Oscillographe** ou **Graphe à balayage**. Le mode par défaut est **Graphe déroulant**. Un graphe déroulant présente des données en cours d'exécution défilant en continu de gauche à droite dans le graphe déroulant. Un oscillographe présente un élément de donnée, comme une impulsion ou une onde, défilant dans le graphe déroulant de gauche à droite. Un graphe à balayage est similaire à un affichage d'électrocardiogramme. Un graphe à balayage fonctionne comme un oscillographe, à l'exception près que le graphe à balayage présente les anciennes données sur la droite et les nouvelles données sur la gauche, séparées par une ligne verticale.

Courbes superposées ou courbes empilées

Vous pouvez afficher plusieurs courbes sur un graphe déroulant en utilisant une échelle verticale unique (courbes "superposées") ou en utilisant plusieurs échelles verticales (courbes "empilées"). La figure 11-2 présente des exemples de courbes superposées et empilées.

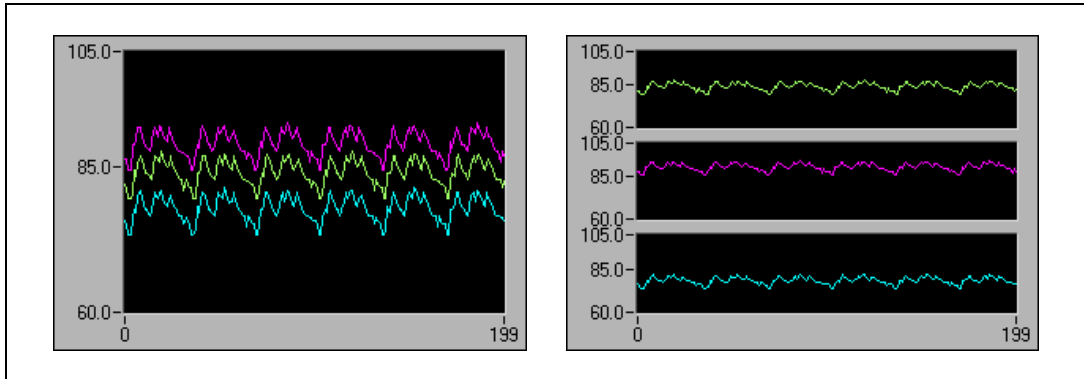


Figure 11-2. Graphes déroulants avec des courbes superposées et empilées

Reportez-vous au VI Charts de la bibliothèque `exemples\general\graphs\charts.llb` pour visualiser des exemples de graphes déroulants différents et des types de données acceptés.

Graphes et graphes XY

Les graphes affichent des mesures échantillonnées régulièrement. Les graphes XY affichent des ensembles de points, échantillonnés régulièrement ou irrégulièrement. La figure 11-3 présente des exemples de graphes et de graphes XY.

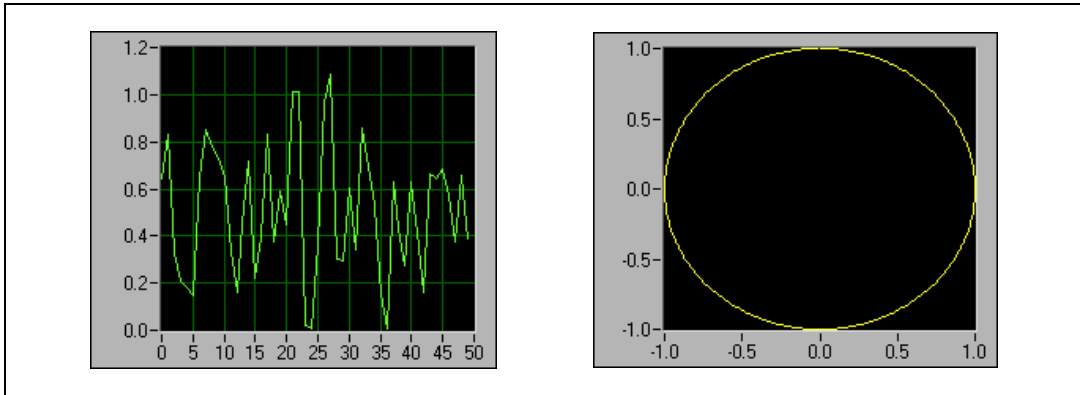


Figure 11-3. Graphes et graphes XY

Le graphe ne trace que les fonctions à valeur unique, comme dans $y = f(x)$, avec des points distribués également le long de l'axe des X, comme les formes d'ondes acquises qui varient dans le temps. Le graphe XY est un objet graphique cartésien à usage général qui trace des fonctions à valeurs multiples, comme des formes circulaires ou des formes d'ondes avec une base de temps qui varie. Les deux graphes affichent des courbes contenant n'importe quel nombre de points.

Ces deux types de graphes acceptent plusieurs types de données, ce qui minimise la manipulation des données avant l'affichage.

Types de données des graphes à courbe unique

Le graphe à courbe unique accepte deux types de données.

Le graphe accepte un tableau de valeurs unique, interprète les données comme des points sur le graphe et incrémente l'indice x d'une unité, en commençant à $x = 0$. Le graphe accepte également un cluster contenant une valeur x initiale, un Δx et un tableau de données y .

Reportez-vous au VI Waveform Graph de la bibliothèque `examples\general\graphs\gengraph.11b` pour visualiser des exemples de types de données acceptés par les graphes à courbe unique.

Graphes multi-courbes

Un graphe à plusieurs courbes accepte un tableau à 2 dimensions (2D), où chaque ligne du tableau est une courbe unique. Le graphe interprète les données comme des points sur le graphe et incrémente l'indice x d'une unité, en commençant à $x = 0$. Câblez un tableau 2D au graphe, cliquez avec le bouton droit sur le graphe et sélectionnez **Transposer le tableau** dans le menu local, pour gérer chaque colonne du tableau comme une courbe. Ceci est particulièrement utile lorsque vous échantillonnez plusieurs voies à partir d'un périphérique DAQ parce que le périphérique retourne les données comme des tableaux 2D, avec chaque voie stockée comme une colonne séparée. Reportez-vous au graphe multi-courbes (Y) Multi Plot 1 du VI Waveform Graph de la bibliothèque `examples\general\graphs\gengraph.11b` pour visualiser un exemple de graphe qui accepte ce type de données.

Un graphe à plusieurs courbes accepte également un cluster contenant une valeur x , une valeur Δx et un tableau 2D de données y . Le graphe interprète les données y comme des points sur le graphe et incrémente l'indice x par Δx , en commençant à $x = 0$. Ce type de données est utile pour afficher plusieurs signaux échantillonnés avec la même régularité. Reportez-vous au graphe multi-courbes (Xo, dX, Y) Multi Plot 3 du VI Waveform Graph de la bibliothèque `examples\general\graphs\gengraph.11b` pour visualiser un exemple de graphe qui accepte ce type de données.

Un graphe multi-courbes accepte un tableau de courbes où le tableau contient des clusters. Chaque cluster contient un tableau de points qui contient les données y . Le tableau intérieur décrit les points dans une courbe et le tableau extérieur comporte un cluster pour chaque courbe. La figure 11-4 présente le tableau de cluster contenant les données y .



Figure 11-4. Tableau de cluster y

Utilisez un graphe multi-courbes au lieu d'un tableau 2D si chaque courbe a un nombre de points différent. Par exemple, lorsque vous échantillonnez des données de voies différentes en utilisant des durées différentes pour chaque voie, utilisez cette structure de données à la place d'un tableau 2D parce que chaque ligne d'un tableau 2D doit avoir le même nombre d'éléments. Le nombre d'éléments dans les tableaux internes d'un tableau de clusters peut varier. Reportez-vous au graphe multi-courbes (Y) Multi Plot 2 du VI Waveform Graph de la bibliothèque `examples\general\graphs\gengraph.llb` pour visualiser un exemple de graphe qui accepte ce type de données.

Un graphe multi-courbes accepte un cluster contenant une valeur x initiale, une valeur Δx et un tableau qui contient des clusters. Chaque cluster contient un tableau de points qui contient les données y . Utilisez la fonction `Assembler` pour assembler les tableaux dans les clusters et utilisez la fonction `Construire un tableau` pour construire le tableau contenant les clusters qui en résultent. Vous pouvez également utiliser la fonction `Construire un tableau de clusters`, qui crée des tableaux de clusters contenant des entrées spécifiées. Reportez-vous au graphe multi-courbes (Xo, dX, Y) Multi Plot 2 du VI Waveform Graph de la bibliothèque `examples\general\graphs\gengraph.llb` pour visualiser un exemple de graphe qui accepte ce type de données.

Un graphe multi-courbes accepte un tableau de clusters contenant une valeur x , une valeur Δx et un tableau de données y . Il s'agit du type de données de graphe multi-courbes le plus général parce que vous pouvez indiquer un point de démarrage unique et incrémenter l'axe des X de chaque courbe. Reportez-vous au graphe multi-courbes (Xo, dX, Y) Multi Plot 1 du VI Waveform Graph de la bibliothèque `examples\general\graphs\gengraph.llb` pour visualiser un exemple de graphe qui accepte ce type de données.

Types de données des graphes XY à courbe unique

Le graphe XY à courbe unique accepte un cluster contenant un tableau x et un tableau y . Le graphe XY accepte également un tableau de points, où un point est un cluster contenant une valeur x et une valeur y .

Reportez-vous au VI XY Graph de la bibliothèque `examples\general\graph\gengraph.llb` pour un exemple de types de données acceptés par les graphes XY à courbe unique.

Types de données des graphes XY multi-courbes

Le graphe XY multi-courbes accepte un tableau de courbes, où une courbe est un cluster contenant un tableau x et un tableau y . Le graphe XY multi-courbes accepte également un tableau de clusters de courbes, où une courbe est un tableau de points. Un point est un cluster contenant une valeur x et une valeur y .

Reportez-vous au VI XY Graph de la bibliothèque `examples\general\graph\gengraph.llb` pour visualiser un exemple de types de données acceptés par les graphes XY multi-courbes.

Graphes déroulants

Le graphe déroulant est un type particulier d'indicateur numérique qui affiche une ou plusieurs courbes. Reportez-vous à la bibliothèque `examples\general\graphs\charts.llb` pour visualiser des exemples de graphes déroulants.

Vous pouvez transmettre au graphe déroulant une ou plusieurs valeurs à la fois. Comme avec le graphe, LabVIEW interprète les données comme des points sur le graphe et incrémente l'indice x d'une unité en commençant à $x = 0$. Le graphe déroulant traite ces entrées comme de nouvelles données pour une courbe unique.

Le graphe déroulant se redessine moins fréquemment lorsque vous lui transmettez plusieurs points à la fois.

Afin de transmettre des données pour plusieurs courbes à un graphe déroulant, vous pouvez assembler les données dans un cluster de valeurs scalaires, où chaque valeur numérique représente un point unique pour chaque courbe.

Si vous voulez transmettre plusieurs points pour des courbes au cours d'une mise à jour unique, câblez un tableau de clusters de valeurs numériques au graphe déroulant. Chaque valeur numérique représente un point de valeur y unique pour chaque courbe.

Si vous ne pouvez pas déterminer le nombre de tracés que vous souhaitez afficher avant l'exécution ou si vous désirez transmettre plusieurs points pour différentes courbes au cours d'une mise à jour unique, câblez un tableau 2D de données au graphe déroulant. Comme pour les graphes, par défaut, les graphes déroulants gèrent les lignes comme de nouvelles données pour chaque courbe. Câblez un type de données de tableau 2D au graphe déroulant, cliquez avec le bouton droit sur le graphe déroulant et

sélectionnez **Transposer le tableau** dans le menu local pour traiter les colonnes dans le tableau comme de nouvelles données pour chaque courbe.

Graphes et graphes déroulants d'intensité

Utilisez le graphe et le graphe déroulant d'intensité pour afficher des données 3D sur un tracé 2D en plaçant des blocs de couleur sur un plan cartésien. Par exemple, vous pouvez utiliser des graphes d'intensité (déroulants ou non) pour afficher une carte de température ou une carte morphographique, dans lesquels l'intensité représente respectivement la température et l'altitude. Le graphe et le graphe déroulant d'intensité acceptent un tableau 2D de nombres. Chaque nombre dans le tableau représente une couleur spécifique. Les indices des éléments dans le tableau 2D définissent l'emplacement des tracés pour les couleurs. La figure 11-5 présente le concept de l'opération pour le graphe déroulant d'intensité.

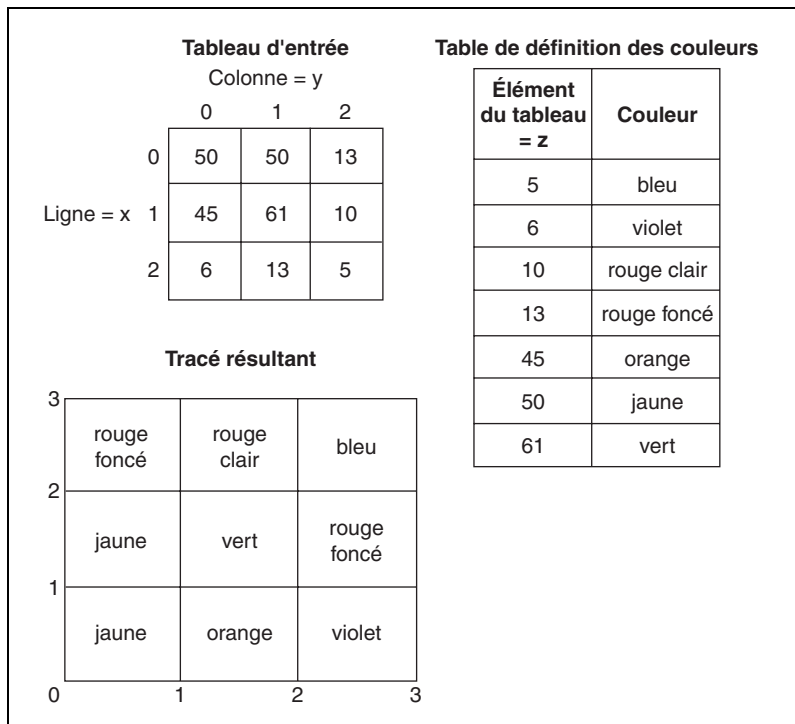


Figure 11-5. Carte des couleurs du graphe déroulant d'intensité

Les lignes de données passent dans l'affichage comme de nouvelles colonnes sur le graphe ou le graphe déroulant. Si vous voulez que les lignes apparaissent comme des lignes sur l'affichage, câblez un type de données de tableau 2D au graphe ou au graphe déroulant, cliquez avec le bouton droit sur le graphe ou le graphe déroulant, puis sélectionnez **Transposer le tableau** dans le menu local.

Les indices du tableau se trouvent en bas à gauche du bloc de couleur. Le bloc de couleur possède une aire d'unité, qui correspond à l'aire entre les deux points, définie par les indices du tableau. Le graphe déroulant d'intensité ou le graphe d'intensité peut afficher jusqu'à 256 couleurs discrètes. Pour afficher de grandes quantités de données sur un graphe déroulant d'intensité, assurez-vous de disposer d'assez de mémoire.

Après avoir positionné un bloc de données sur un graphe déroulant d'intensité, l'origine du plan cartésien se décale vers la droite du dernier bloc de données. Lorsque le graphe déroulant traite de nouvelles données, celles-ci apparaissent à droite des anciennes données. Lorsque l'affichage d'un graphe déroulant est plein, les données les plus anciennes débordent à gauche du graphe déroulant. Ce comportement est similaire au comportement des graphes déroulants. Reportez-vous à la section [Mode de mise à jour du graphe déroulant](#) de ce chapitre, pour obtenir plus d'informations sur ces graphes déroulants.

Reportez-vous à la bibliothèque `examples\general\graphs\intgraph.llb` pour visualiser des exemples de graphes déroulants d'intensité et de graphes d'intensité.

Représentation des couleurs

Vous pouvez définir interactivement la représentation des couleurs pour des graphes d'intensité et des graphes déroulants d'intensité, de la même façon que vous définissez les couleurs pour une commande numérique de rampe de couleurs. Reportez-vous à la section [Rampes de couleurs](#) du chapitre 4, [Construction de la face-avant](#), pour obtenir plus d'informations sur les rampes de couleurs.

Vous pouvez définir la représentation des couleurs du graphe d'intensité et du graphe déroulant d'intensité par programmation en utilisant le nœud de propriété de deux façons. Habituellement, vous spécifiez la représentation de la couleur dans le nœud de propriété. Pour cette méthode, spécifiez la propriété **Échelle Z : Valeurs du repère**. Cette propriété consiste en un tableau de clusters, dans lequel chaque cluster contient une valeur numérique limite avec la couleur correspondante à afficher pour cette valeur. Lorsque vous spécifiez la représentation de couleur de cette

manière, vous pouvez spécifier une couleur hors gamme supérieure dans la propriété **Échelle Z : Couleur sup.** et une couleur hors gamme inférieure en utilisant la propriété **Échelle Z : Couleur inf.** Le graphe d'intensité et le graphe déroulant d'intensité sont limités à 254 couleurs, avec les couleurs hors gamme inférieure et supérieure portant le total à 256 couleurs. Si vous spécifiez plus de 254 couleurs, le graphe d'intensité ou le graphe déroulant d'intensité crée la table de 254 couleurs en interpolant parmi les couleurs spécifiées.

Si vous affichez un bitmap sur le graphe d'intensité, vous spécifiez une table de couleurs en utilisant la propriété **Table de couleurs**. Avec cette méthode, vous pouvez spécifier un tableau de 256 couleurs maximum. Les données transmises au graphe déroulant sont reliées aux indices dans cette table de couleurs sur la base de l'échelle de couleurs du graphe déroulant d'intensité. Si l'échelle de couleurs est comprise entre 0 et 100, une valeur de 0 dans les données est reliée à l'indice 1 et une valeur de 100 est reliée à l'indice 254, avec des valeurs intérieures interpolées entre 1 et 254. Toutes les données au-dessous de 0 sont reliées à la couleur hors de la gamme inférieure (indice 0) et toutes les données au-dessus de 100 sont reliées à la couleur hors de la gamme supérieure (indice 255).



Remarque Les couleurs que vous voulez que le graphe d'intensité ou le graphe déroulant d'intensité affiche sont limitées aux couleurs exactes et aux nombres de couleurs supportées par votre carte vidéo. Vous êtes également limité par le nombre de couleurs de l'affichage.

Options du graphe déroulant d'intensité

Le graphe déroulant d'intensité et les graphes déroulants ont de nombreuses options en commun, que vous pouvez afficher ou masquer en cliquant avec le bouton droit sur le graphe déroulant et en sélectionnant **Éléments visibles** dans le menu local. En outre, étant donné que le graphe déroulant d'intensité inclut des couleurs sous la forme d'une troisième dimension, une échelle semblable à une commande de rampe de couleurs définit la gamme et la correspondance des valeurs aux couleurs.

Comme les graphes déroulants, le graphe déroulant d'intensité conserve un historique des données, ou buffer, à partir des mises à jour précédentes. Vous pouvez configurer le buffer en cliquant avec le bouton droit sur le graphe déroulant et en sélectionnant **Longueur de l'historique dans le menu local**. La taille par défaut d'un graphe déroulant d'intensité est de 128 points de données. L'affichage du graphe déroulant d'intensité peut utiliser beaucoup de mémoire. Par exemple, afficher un graphe déroulant

à simple précision avec un historique de 512 points et 128 valeurs de y nécessite $512 * 128 * 4$ octets (la taille d'un nombre à simple précision) ou 256 Ko.

Options du graphe d'intensité

Le graphe d'intensité fonctionne comme le graphe déroulant d'intensité, sauf qu'il ne retient pas les données précédentes et qu'il n'inclut pas de modes de rafraîchissement. Chaque fois que de nouvelles données passent dans un graphe d'intensité, elles remplacent les anciennes données.

Le graphe d'intensité peut avoir des curseurs comme les autres graphes. Chaque curseur affiche les valeurs x , y et z pour un point spécifié sur le graphe.

Graphes numériques

Utilisez un graphe pour afficher des données numériques, surtout lorsque vous utilisez des diagrammes temporels ou des analyseurs logiques. Reportez-vous au chapitre 8, *Digital I/O*, du *LabVIEW Measurements Manual* pour obtenir plus de détails sur l'acquisition de données numériques.

Le graphe de la figure 11-6 affiche un tableau de six entiers 8 bits non signés. Le graphe représente les entiers au format binaire, et chaque bit représente une courbe sur le graphe.

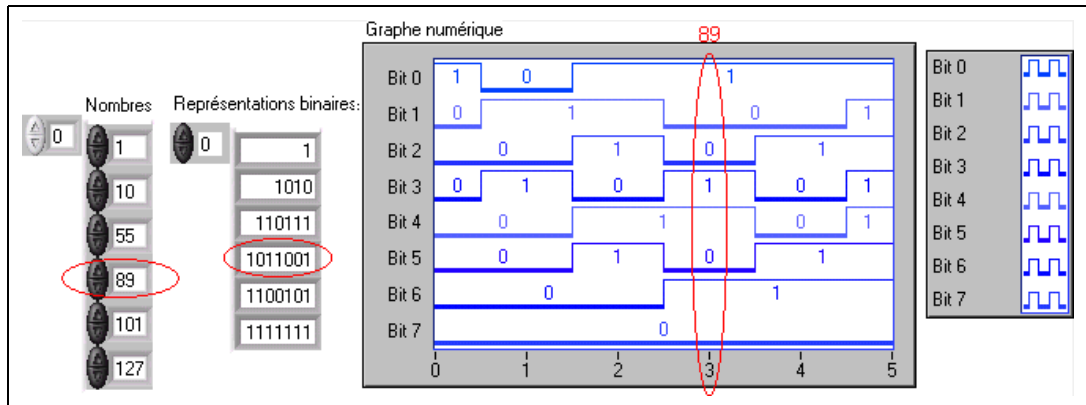


Figure 11-6. Représentation graphique d'entiers sous forme numérique

La colonne **Représentations binaires** dans la figure 11-6 correspond à la première courbe, Bit 0, sur le graphe numérique. La colonne **Nombres** dans la figure 11-6 correspond à la deuxième courbe, Bit 1, etc. Par exemple, le nombre 89 nécessite 7 bits de mémoire. La représentation graphique binaire du chiffre 89 apparaît au point 3 dans le graphe.

Pour construire un VI qui affiche des données numériques, utilisez la fonction **Assembler** pour assembler l'instant du déclenchement (X0), Δx (dX), le tableau de nombres et le nombre de ports, comme illustré dans la figure 11-7.

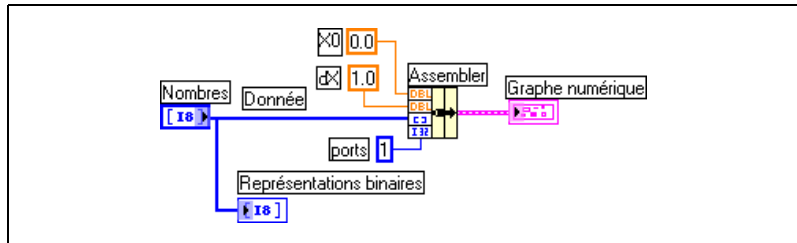


Figure 11-7. Utilisation de la fonction **Assembler** avec un graphe numérique

ports spécifie le nombre d'éléments de données à traiter comme un entier unique. Si vous avez des données 8 bits représentées par un entier 8 bits, le nombre de ports sera égal à 1. Si vous avez des données 32 bits représentées par un entier 32 bits, le nombre de ports sera aussi de 1. Si vous avez trois sources de données 8 bits, vous aurez 24 bits à représenter par un entier unique. Dans ce cas, utilisez la fonction **Entrelacer** pour entrelacer les données 8 bits et spécifier un nombre de ports de 3, comme illustré dans la figure 11-8.

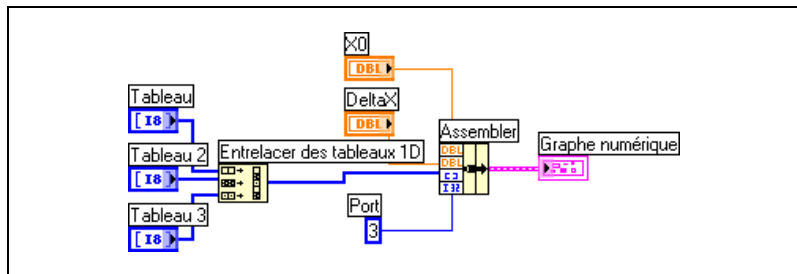


Figure 11-8. Utilisation de la fonction **Entrelacer** avec un graphe numérique

Dans la figure 11-8, entrez 3 dans **Port** pour représenter les trois tableaux de données 8 bits sous forme de graphique. Entrez 2 dans **Port** pour représenter deux tableaux de données 8 bits sous forme de graphique.

Entrez 5 dans **Port** pour représenter cinq tableaux de données 8 bits sous forme de graphique. À la suite de cette action, 16 des 40 bits ne vont contenir aucune valeur.

Masquage des données

Le VI de la figure 11-6 fournit un graphe dans lequel chaque courbe représente un bit des données. Vous pouvez aussi sélectionner, réorganiser et concaténer des bits dans les données avant de les afficher dans le graphe. L'opération qui consiste à sélectionner, réorganiser et concaténer des bits est appelée “masquage des données”.

Utilisez un masque pour concaténer les courbes d'au moins deux bits différents et pour les afficher sous forme de courbe unique. Si vous avez un tableau d'entiers 8 bits, vous pouvez assembler jusqu'à 8 bits pour les afficher sur une courbe unique. Si vous avez un tableau d'entiers 16 bits, vous pouvez afficher jusqu'à 16 bits sur une courbe unique, et ainsi de suite. Vous pouvez aussi tracer le même bit deux fois ou plus sur une courbe unique. Reportez-vous à l'annexe D, *Masquage des données numériques*, pour plus de détails sur le masquage de données numériques.

Graphes 3D



Remarque Les commandes de graphe 3D ne sont disponibles que pour Windows dans les versions Systèmes de développement complet et professionnel de LabVIEW.

Pour de nombreux ensembles de données réelles, comme la répartition de la température sur une surface, l'analyse temps-fréquence et la trajectoire d'un avion, il vous faut visualiser des données en trois dimensions. Vous pouvez visualiser des données en trois dimensions à l'aide d'un graphe 3D et changer l'affichage de ces données en modifiant les propriétés du graphe 3D.

Les graphes 3D disponibles sont les suivants :

- **Graphe 3D de surface** : dessine une surface dans l'espace 3D. Lorsque vous placez cette commande sur la face-avant, LabVIEW la relie à un sous-VI qui reçoit les données représentant la surface.
- **Graphe 3D Paramétrique** : dessine une surface complexe dans l'espace 3D. Lorsque vous placez cette commande sur la face-avant, LabVIEW la relie à un sous-VI qui reçoit les données représentant la surface.

- **Graphe 3D de courbe** : trace une ligne dans l'espace 3D. Lorsque vous placez cette commande sur la face-avant, LabVIEW la relie à un sous-VI qui reçoit les données représentant la ligne.

Utilisez les graphes 3D conjointement avec les VIs de graphes 3D situés dans la palette **Fonctions»Graphisme et Son»Propriétés de graphe 3D** pour tracer des courbes et des surfaces. Une courbe est composée de points distincts sur le graphe, qui ont chacun une coordonnée x , y , et z . Le VI relie ces points à une ligne. Une courbe est idéale pour visualiser le chemin d'un objet en mouvement, par exemple la trajectoire de vol d'un avion.

Les graphes 3D utilisent la technologie ActiveX et les VIs qui gèrent la représentation 3D. Vous pouvez définir des propriétés de manière à ce que les VIs situés dans la palette **Fonctions»Graphisme et Son»Propriétés de graphe 3D** modifient leur comportement lors de l'exécution. Vous pouvez ainsi définir les propriétés élémentaires, les propriétés des axes, les propriétés de grille et les propriétés de projection.

Une courbe de surface utilise des données x , y , et z pour tracer des points sur le graphe. La courbe de surface relie ces points et forme ainsi une vue de la surface des données en trois dimensions. Ainsi, vous pouvez utiliser une courbe de surface pour effectuer un relevé de terrain.

Lorsque vous sélectionnez un graphe 3D, LabVIEW place un conteneur ActiveX sur la face-avant qui contient une commande de graphe 3D. LabVIEW place aussi une référence à la commande de graphe 3D dans le diagramme. LabVIEW relie cette référence à l'un des trois VIs de graphe 3D.

Type de données waveform

Le type de données waveform est un cluster d'éléments qui comporte les données, le temps de départ et le Δt d'un signal. Vous pouvez créer des waveforms à l'aide de la fonction **Construire une waveform** située dans la palette **Fonctions»Waveform**. Un grand nombre des VIs et des fonctions que vous utilisez pour acquérir et analyser les signaux acceptent et retournent le type de données waveform par défaut. Lorsque vous câblez des données de type waveform à un graphe ou à un graphe déroulant, le graphe ou le graphe déroulant trace automatiquement une courbe (waveform) basée sur les données, le temps de départ et le Δt du signal. Lorsque vous câblez un tableau de données de type waveform à un graphe ou à un graphe déroulant, le graphe ou le graphe déroulant trace automatiquement toutes les courbes. Reportez-vous au chapitre 5, *Introduction to Data Acquisition in LabVIEW*, du *LabVIEW Measurements Manual* pour plus d'informations sur l'utilisation du type de données waveform.

VIs Graphisme et son

Utilisez les VIs et les fonctions graphiques et audio de la palette **Fonctions»Graphisme et son** pour afficher ou modifier des graphiques et des sons dans les VIs.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour plus d'informations sur l'utilisation de graphiques et de sons dans les VIs.

Utilisation de l'indicateur image

L'indicateur image, situé dans la palette **Commandes»Graphe»Cmds**, est un indicateur à usage général pour l'affichage d'images contenant des lignes, des cercles, du texte et d'autres types graphiques. Comme vous avez un contrôle au niveau des pixels sur l'indicateur image, vous pouvez créer presque tous les objets graphiques que vous désirez.

L'indicateur image dispose d'un système de coordonnées basé sur les pixels, dans lequel l'origine (0,0) correspond au pixel situé à l'angle supérieur gauche de la commande. La composante (x) horizontale d'une coordonnée croît vers la droite, tandis que la coordonnée (y) verticale croît vers le bas.

Si une image est trop grande pour l'indicateur image qui l'affiche, LabVIEW découpe l'image pour que seuls les pixels de la zone d'affichage de l'indicateur s'affichent. Utilisez l'outil Flèche pour redimensionner l'indicateur et exécutez le VI à nouveau afin d'afficher l'image toute entière.

Pour lire ou changer par programmation la taille de la zone d'image de l'indicateur image, utilisez la propriété **Taille zone de dessin**. Pour lire la taille de l'indicateur image en entier, utilisez la propriété **Limites**. Vous pouvez également utiliser l'outil Flèche pour sélectionner l'indicateur, sélectionner **Édition»Personnaliser la commande...** et utiliser la fenêtre **Composantes de la commande** de l'Éditeur de commande pour déterminer la taille de l'indicateur image en entier et la taille de la région d'affichage de l'indicateur image.



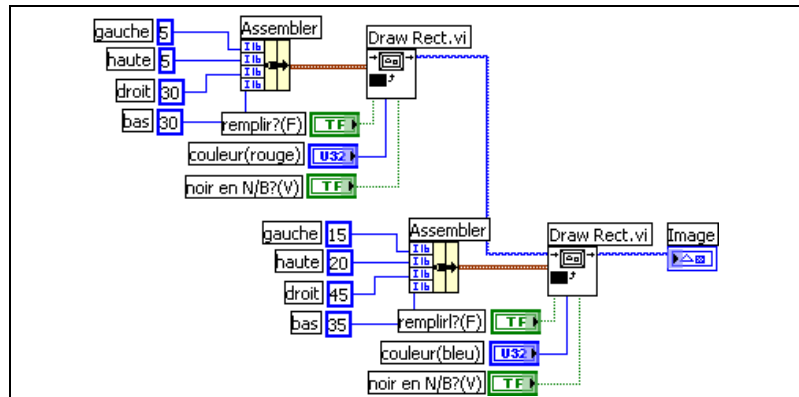
Lorsque vous placez un indicateur image sur la face-avant, il apparaît sous la forme d'une zone rectangulaire vierge et un terminal correspondant, présenté sur la gauche, apparaît sur le diagramme. Utilisez les VIs de fonctions sur les images, situés sur la palette **Fonctions»Graphisme et son»Fonctions sur les images** au lieu d'une application pour modifier et ajouter des fonctionnalités aux graphiques dans LabVIEW.

Pour afficher une image dans un indicateur image, vous devez utiliser les VIs de fonctions sur les images pour spécifier un ensemble d'instructions graphiques. Chaque VI correspond à un ensemble d'entrées décrivant une instruction graphique. Sur la base de ces entrées, le VI crée une description compacte de ces spécifications que vous transmettez à l'indicateur image pour l'affichage.

Vous ne pouvez connecter les images créées avec les VIs de fonctions sur les images qu'à un indicateur image ou à l'entrée **image** d'un VI de fonctions sur les images. LabVIEW dessine l'image lorsqu'il met à jour l'indicateur image sur une face-avant ouverte.

Chaque VI de fonctions sur les images enchaîne les instructions graphiques de l'**image** d'entrée à l'**image** de sortie. Si l'entrée de l'**image** n'est pas câblée, la sortie de l'**image** retourne une zone rectangulaire vierge.

Le diagramme suivant utilise le VI Draw Rect pour tracer deux rectangles qui se chevauchent.



La commande booléenne **noir en N/B ?** indique si le rectangle est noir ou blanc et l'affiche sur un moniteur monochrome. Si vous créez un jeu de VIs qui utilisent l'indicateur image et que vous souhaitez que les VIs s'exécutent sur tous les moniteurs, vous devez utiliser la commande **noir en N/B ?**.

VIs de tracés d'images

Utilisez les VIs de tracés d'images, situés sur la palette **Fonctions» Graphisme et son»Tracés images**, pour créer des types de graphes courants en utilisant l'indicateur image. Ces graphes comprennent un tracé polaire, un affichage de graphe, un tracé Smith et une échelle de graphe.

Le VI Tracé images utilise des fonctions graphiques de bas niveau pour créer un affichage graphique de vos données et personnaliser le code de tracé pour ajouter des fonctionnalités. Ces affichages graphiques ne sont pas aussi interactifs que les commandes intégrées de LabVIEW, mais vous pouvez les utiliser pour visualiser des informations que les commandes intégrées ne permettent pas. Utilisez le VI Plot Waveform pour créer un tracé présentant des fonctionnalités légèrement différentes que celles des graphes intégrés de LabVIEW.

Utilisation du VI Polar Plot comme sous-VI

Utilisez le VI Polar Plot pour dessiner des quadrants contigus spécifiques d'un graphe polaire ou le graphe tout entier. Comme avec les graphes intégrés de LabVIEW, vous pouvez spécifier la couleur des composantes, inclure une grille, et spécifier la gamme et le format des échelles.

Le VI Polar Plot fournit de nombreuses fonctionnalités dans un VI unique. Par conséquent, le VI inclut des clusters complexes comme entrées. Vous pouvez utiliser les valeurs par défaut et les commandes personnalisées pour diminuer la complexité du VI. Au lieu de créer votre propre entrée de cluster sur le diagramme, copiez une commande personnalisée à partir du VI Polar Plot Demo situé dans la bibliothèque `examples\picture\demos.llb` et placez-la sur votre face-avant.

Utilisation du VI Plot Waveform comme sous-VI

Utilisez le VI Plot Waveform, qui simule le comportement du graphe intégré de LabVIEW, pour tracer des waveforms de styles variés, y compris des points, des lignes connectées et des barres. Comme avec les graphes waveform intégrés de LabVIEW, vous pouvez spécifier la couleur des composantes, inclure une grille, et spécifier la gamme et le format des échelles.

Le VI Waveform Plot fournit de nombreuses fonctionnalités dans un VI unique. Par conséquent, le VI inclut des clusters complexes comme entrées. Vous pouvez utiliser les valeurs par défaut et les commandes personnalisées pour diminuer la complexité du VI. Au lieu de créer votre propre entrée de

cluster, copiez une commande personnalisée du VI Waveform and XY Plots de la bibliothèque `examples\picture\demos.llb` et mettez-la sur votre face-avant.

Le VI Plot XY et le VI Plot Multi-XY sont semblables au VI Plot Waveform. Vous utilisez des commandes différentes pour spécifier l'apparence du tracé parce que les VIs Plot XY possèdent trois styles de courbe : deux styles de courbe de dispersion et un style de courbe où une ligne est tracée à chaque position x unique pour marquer les valeurs de y maximales et minimales pour cette valeur de x .

Utilisation des VIs Smith Plot comme sous-VIs

Utilisez les VIs Smith Plot pour étudier le comportement des lignes de transmission, comme dans l'industrie des télécommunications. Une ligne de transmission est un support à travers lequel vous transmettez de l'énergie et des signaux. Il peut s'agir d'un fil de liaison ou de l'atmosphère dans laquelle un signal est transmis. Les lignes de transmission ont un effet sur le signal qui est transmis. Cet effet, appelé l'impédance de la ligne de transmission, peut atténuer ou décaler la phase d'un signal AC.

L'impédance de la ligne de transmission est une mesure de la résistance et de la réactance de la ligne. L'impédance, z , est couramment désignée par un nombre complexe sous la forme $z = r + jx$, où la résistance (r) et la réactance (x) sont des composantes.

Utilisez les VIs Smith Plot pour afficher les impédances de lignes de transmission. Le tracé est constitué de cercles de résistance et de réactance constantes.

Vous pouvez tracer une impédance donnée, $r + jx$, en repérant l'intersection du cercle r et du cercle x appropriés. Une fois que vous avez tracé l'impédance, utilisez les VIs Smith Plot comme aide visuelle pour concorder avec l'impédance et pour calculer le coefficient de réflexion d'une ligne de transmission.

Les VIs Smith Plot fournissent de nombreuses fonctionnalités dans un VI unique. C'est pourquoi la plupart de ces VIs comportent en entrées des clusters complexes. Vous pouvez utiliser les valeurs par défaut et les commandes personnalisées pour diminuer la complexité du VI. Au lieu de créer votre propre entrée de cluster, copiez une commande personnalisée du VI Waveform and XY Plots dans la bibliothèque `examples\picture\demos.llb` et placez-la sur votre face-avant.

Si vous tracez des impédances de charge, vous pouvez représenter une impédance comme un nombre complexe sous la forme $r + jx$.

Pour éviter de perdre les détails de la courbe Smith, utilisez le VI Normalize Smith Plot pour normaliser les données. Vous pouvez transmettre les données que vous normalisez avec le VI Normalize Smith Plot directement au VI Smith Plot. Vous mettez généralement à l'échelle les données de la courbe Smith en respectant l'impédance caractéristique (Z_0) du système.

VIs Fonctions sur les images

Utilisez les VIs Fonctions sur les images, situés sur la palette **Fonctions»Graphisme et son»Fonctions sur les images**, pour dessiner des formes et entrer du texte dans un indicateur image. Vous pouvez tracer des points, des lignes, des formes et des tables de pixels. Les tables de pixels de données non aplaties correspondent à des tableaux 2D de couleurs, où chaque valeur correspond à une couleur.

La première rangée de la palette **Fonctions sur les images** contient des VIs que vous utilisez pour tracer des points et des lignes. Un point est un cluster de deux entiers 16 bits signés représentant les coordonnées x et y d'un pixel. Lorsque vous dessinez, l'image se souvient de la position du marqueur graphique.

Pour la plupart des VIs Fonctions sur les images, vous devez spécifier des coordonnées absolues, c'est-à-dire relatives à l'origine (0,0). Avec le VI Draw Line et le VI Move Pen, vous pouvez spécifier des coordonnées absolues ou relatives. Des coordonnées relatives se réfèrent à la position actuelle du marqueur. Vous pouvez utiliser le VI Move Pen pour changer la position du marqueur sans dessiner. Seuls les VIs Draw Point, Move Pen, Draw Line et Draw Multiple Lines changent la position du marqueur.

La seconde rangée de la palette **Fonctions sur les images** contient des VIs que vous utilisez pour tracer une forme remplie. Chacun de ces VIs trace une forme dans la zone rectangulaire d'une image. Vous spécifiez un rectangle comme un cluster de quatre valeurs qui représentent les pixels gauche, haut, droit et bas. Avec le VI Draw Arc, le rectangle décrit la taille d'un ovale. Des paramètres supplémentaires spécifient la portion de l'ovale (l'arc) que vous désirez dessiner.

La troisième rangée de la palette **Fonctions sur les images** contient des VIs que vous utilisez pour tracer du texte dans une image. Le VI Get Text Rect ne trace pas du texte. Vous l'utilisez pour calculer la taille d'un rectangle de délimitation d'une chaîne.

La quatrième rangée de la palette **Fonctions sur les images** contient des VIs que vous utilisez pour tracer des tables de pixels dans une image. En plus de spécifier le tableau 2D de données, vous spécifiez également la table de couleurs pour les valeurs de tableau correspondant aux couleurs. Le tableau 2D correspond à une grille de pixels. Ces VIs convertissent chaque valeur dans le tableau de données en une couleur en utilisant la valeur comme indice dans le tableau de la table de couleurs.

La dernière rangée de la palette **Fonctions sur les images** contient la constante Image vide, que vous utilisez chaque fois que vous démarrez avec une image vide ou que vous effectuez des changements sur une image vide.

Création et modification des couleurs avec les VIs Fonctions sur les images

De nombreux VIs Fonctions sur les images possèdent une entrée **couleur** pour modifier la couleur des formes et du texte. Pour créer une couleur, le plus simple est d'utiliser une constante boîte de couleurs, située sur la palette **Fonctions»Numérique»Constantes numériques supplémentaires** et de cliquer sur la constante pour sélectionner une couleur.

Pour créer des couleurs à la suite de calculs plutôt qu'en utilisant des constantes boîte de couleurs, vous devez comprendre comment une boîte de couleurs spécifie une couleur avec une valeur numérique.

Un entier 32 bits signé représente une couleur, et les trois octets inférieurs représentent les composantes rouge, verte et bleue de la couleur. Pour une gamme de couleurs bleues, créez un tableau d'entiers 32 bits où seules les valeurs des octets inférieurs changent parce que l'octet inférieur contient la composante bleue. Pour une gamme de couleurs grises, créez un tableau d'entiers 32 bits où les valeurs rouge, vert et bleu de chaque élément sont les mêmes.

VIs Formats graphiques

Utilisez les VIs Formats graphiques, situés sur la palette **Fonctions»Graphisme et son»Formats graphiques**, pour lire et écrire des données dans des fichiers de différents formats graphiques standard, y compris des fichiers bitmap (.bmp), Portable Network Graphics (.png) et Joint Photographic Expert Group (.jpg). Vous pouvez utiliser ces VIs pour accomplir les tâches suivantes :

- Lire des données d'un fichier bitmap pour les afficher dans un indicateur de type image.
- Lire des données pour la manipulation d'image.
- Écrire une image pour l'afficher dans d'autres applications.

Les données bitmap correspondent à des tableaux 2D dans lesquels le type de données de chaque point varie selon la profondeur de couleurs. Par exemple, dans une image noir et blanc ou 1 bit, chaque point est booléen. Dans des images 4 bits et 8 bits, chaque point est un indice dans une table de couleurs. Pour des images 24 bits en mode vraies couleurs, chaque point est composé de différentes valeurs de rouge, vert et bleu (RVB).

Les VIs qui lisent et écrivent des fichiers graphiques fonctionnent avec des données dans un format simple et aplati, plus proche de la manière dont les fichiers sont écrits sur un disque, avec les données stockées dans un tableau 1D. Ces fichiers graphiques sont des tables de pixels, dont le concept est proche de celui des bitmaps. Vous pouvez afficher ces données aplaties en utilisant directement le VI Draw Flattened Pixmap, situé sur la palette **Fonctions»Graphisme et son»Fonctions sur les images**. Cette palette contient des VIs pour le traçage de tables de pixels constituées de tableaux 2D 1 bit (noir et blanc), 4 bits, 8 bits et 24 bits.

Pour manipuler les données sous forme d'un tableau 2D, vous pouvez les convertir au format approprié en utilisant le VI Unflatten Pixmap et le VI Flatten Pixmap situé sur la palette **Fonctions»Graphisme et son»Formats graphiques**.

VIs Son

Utilisez les VIs Son, situés sur la palette **Fonctions»Graphisme et Son» Son**, pour intégrer des fichiers et des fonctions audio dans vos VIs. Vous pouvez utiliser ces VIs pour accomplir les tâches suivantes :

- Créer des VIs lisant des fichiers audio, comme une mise en garde enregistrée, lorsque les utilisateurs réalisent certaines actions.
- Créer un VI lisant un fichier audio lorsqu'il démarre ou termine l'exécution, ou lorsque vous atteignez un certain point dans le VI.
- Configurer un périphérique d'entrée audio pour acquérir des données sonores. Utilisez les VIs Entrée de son pour acquérir les données sonores. Vous pouvez également lire des informations audio dans le périphérique.
- Configurer un périphérique de sortie de son pour accepter des données sonores d'autres VIs Son. Vous pouvez contrôler le volume sonore dans le périphérique, lire ou mettre sur pause le fichier audio ou le supprimer de votre système.

E/S sur fichiers

Les opérations d'E/S sur fichiers transfèrent des données à des fichiers et depuis des fichiers. Utilisez les VIs et fonctions d'E/S sur fichiers, qui se trouvent dans la palette **Fonctions»E/S sur fichiers**, pour gérer tous les aspects d'E/S sur fichiers, y compris les opérations suivantes :

- Ouverture et fermeture de fichiers de données.
- Lecture de données à partir de fichiers et écriture de données dans des fichiers.
- Lecture et écriture dans des fichiers au format tableur.
- Déplacer et renommer des fichiers et des répertoires.
- Modification des caractéristiques d'un fichier.
- Création, modification et lecture d'un fichier de configuration.

Utilisez les VIs de haut niveau pour effectuer les opérations courantes d'E/S. Utilisez les VIs et fonctions de bas niveau pour contrôler individuellement chaque opération d'E/S sur fichiers.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir plus de détails sur les opérations d'E/S sur fichiers.

Opérations de base d'E/S sur fichiers

Pour effectuer une opération d'E/S sur fichiers, procédez comme suit :

1. Créez ou ouvrez un fichier. Indiquez où se trouve un fichier existant ou créez un nouveau fichier en spécifiant un chemin ou en répondant à une boîte de dialogue pour guider LabVIEW vers l'emplacement du fichier. Lorsque le fichier est ouvert, un refnum le représente. Reportez-vous à la section *Références à des objets ou à des applications* du chapitre 4, *Construction de la face-avant*, pour obtenir plus de détails sur les refnums.
2. Lisez ou écrivez dans le fichier.
3. Fermez le fichier.

La plupart des VIs et des fonctions d'E/S sur fichiers n'effectuent qu'une seule étape des opérations d'E/S sur fichiers. Cependant, certains VIs d'E/S sur fichiers de haut niveau, conçus pour effectuer des opérations courantes d'E/S sur fichiers, effectuent chacune des trois étapes. Ces VIs ne sont pas toujours aussi performants que les fonctions de bas niveau, mais ils sont plus faciles à utiliser.

Choix du format d'E/S sur fichiers

Les VIs d'E/S sur fichiers que vous utilisez sont fonction du format des fichiers. Vous pouvez lire ou écrire des données dans des fichiers dans trois formats : texte, binaire et journal. Le format que vous utilisez dépend des données que vous allez acquérir ou créer et des applications qui vont avoir accès à ces données.

Utilisez les indications de base suivantes pour déterminer quel format utiliser :

- Si vous voulez rendre vos données accessibles à d'autres applications, comme par exemple Excel, utilisez des fichiers texte qui sont les fichiers les plus courants et les plus portables.
- Si vous avez besoin d'un accès aléatoire à des lectures ou des écritures et si la vitesse et l'espace disque compact sont essentiels, utilisez des fichiers binaires qui sont plus performants que les fichiers texte en termes d'espace disque et de rapidité.
- Si vous voulez manipuler des enregistrements de données complexes ou des types de données différents dans LabVIEW, utilisez les fichiers journaux qui représentent le meilleur moyen de stocker des données si vous avez l'intention de n'accéder à ces dernières qu'à partir de LabVIEW et si vous avez besoin de stocker des structures de données complexes.

Quand utiliser des fichiers texte

Utilisez des fichiers texte afin de rendre vos données accessibles à d'autres utilisateurs ou à d'autres applications, si l'espace disque et la vitesse d'E/S sur fichiers ne sont pas essentiels, si vous ne devez pas effectuer de lectures ou d'écritures en accès aléatoire et si la précision numérique n'est pas importante.

Les fichiers texte sont le format le plus facile à utiliser et à partager. Pratiquement tous les ordinateurs peuvent lire ou écrire dans un fichier texte. Un grand nombre de programmes textuels peuvent lire les fichiers

texte. La plupart des applications de contrôle d'instruments utilisent des chaînes texte.

Stockez des données dans des fichiers texte lorsque vous voulez pouvoir y accéder à partir d'autres applications, comme par exemple d'une application de traitement de texte ou de tableur. Pour stocker des données en format texte, utilisez les fonctions chaîne situées dans la palette **Fonctions»Chaîne** afin de convertir toutes les données en chaînes texte. Les fichiers texte peuvent contenir des informations de différents types de données.

Les fichiers texte occupent généralement plus de mémoire que les fichiers binaires et que les fichiers journaux si les données initiales ne sont pas au format texte comme le sont, par exemple, les données de graphe ou de graphe déroulant. En effet, la représentation ASCII des données est habituellement plus grande que les données elles-mêmes. Ainsi, vous pouvez stocker le nombre -123,4567 dans 4 octets en tant que flottant simple précision. Cependant, sa représentation ASCII occupera 9 octets, c'est-à-dire un octet par caractère.

De plus, il est difficile d'accéder de façon aléatoire à des données numériques dans des fichiers texte. Bien que chacun des caractères d'une chaîne occupe exactement l'espace d'1 octet, l'espace nécessaire pour exprimer un nombre en texte n'est généralement pas fixe. Pour rechercher le neuvième nombre d'un fichier texte, LabVIEW doit d'abord lire et convertir les huit nombres précédents.

Vous pouvez perdre en précision si vous stockez des données numériques dans des fichiers texte. Les ordinateurs stockent les données numériques sous forme de données binaires, et vous écrivez en général en notation décimale des données numériques dans un fichier texte. Une perte de précision peut se produire lorsque vous lisez des données à partir du fichier texte. Le problème de la perte de précision ne se pose pas avec les fichiers binaires.

Reportez-vous à `examples\file\smpfile.llb` et à `examples\file\sprdsht.llb` si vous souhaitez consulter des exemples d'utilisation d'E/S sur fichiers texte.

Quand utiliser des fichiers binaires

Un fichier binaire utilise un nombre fixe d'octets de stockage sur le disque. Ainsi, le stockage de tout nombre de 0 à 4 milliards au format binaire, comme par exemple 1, 1 000, et 1 000 000, occupe 4 octets par nombre.

Utilisez les fichiers binaires pour enregistrer des données numériques et pour accéder à des nombres particuliers ou accéder de façon aléatoire à des nombres à partir d'un fichier. Les fichiers binaires ne peuvent être lus que par une machine, à la différence des fichiers texte. Les fichiers binaires représentent le format de stockage des données le plus compact et le plus rapide. Vous pouvez utiliser plusieurs types de données différents dans des fichiers binaires, mais cette utilisation est peu fréquente.

Les fichiers binaires sont plus performants, parce qu'ils utilisent moins d'espace disque et parce qu'il n'est pas nécessaire de convertir des données en ou depuis un fichier texte lorsque vous stockez et que vous récupérez des données. Un fichier binaire peut représenter 256 valeurs différentes dans un espace disque d'1 octet. Les fichiers binaires contiennent souvent une image octet-pour-octet des données telles qu'elles ont été stockées en mémoire, à l'exception d'un certain nombre de cas, comme les numériques étendus et les numériques complexes. Lorsque le fichier contient une image octet-pour-octet des données telles qu'elles ont été stockées en mémoire, la lecture du fichier est plus rapide, puisqu'il n'est pas nécessaire d'effectuer une conversion. Reportez-vous à la note d'application *LabVIEW Data Storage* pour plus d'informations sur la façon dont LabVIEW stocke les données.



Remarque Les fichiers texte et les fichiers binaires sont connus sous le nom de fichiers standard, ce qui signifie qu'ils stockent les données comme une séquence de caractères ou d'octets.

Reportez-vous aux VIs Read Binary File et Write Binary File situés dans `examples\file\smpfile.llb` si vous voulez consulter des exemples de lecture et d'écriture d'un tableau de valeurs à virgule flottante double précision, respectivement vers et à partir d'un fichier binaire.

Quand utiliser des fichiers journaux

Utilisez les fichiers journaux pour accéder à des données et les manipuler uniquement dans LabVIEW et si vous avez besoin de stocker des structures de données complexes rapidement et facilement.

Les fichiers journaux sont des fichiers qui ne peuvent être créés et lus que par LabVIEW. À la différence des fichiers binaires qui ne peuvent stocker que des types de données similaires, vous pouvez utiliser les fichiers journaux pour stocker plusieurs types différents de données dans un seul fichier. Ainsi, vous pouvez stocker des chaînes, des numériques et des clusters dans le même fichier journal.

Un fichier journal stocke des données sous la forme d'une séquence d'enregistrements de structures identiques, comme un fichier tableur dans lequel chaque ligne représente un enregistrement. Chaque enregistrement dans un fichier journal doit être du type de données qui lui sont associées. LabVIEW écrit chaque enregistrement dans le fichier comme un cluster contenant les données à stocker. Cependant les composants d'un enregistrement de données peuvent être de tout type de données, que vous déterminez lorsque vous créez le fichier.

Par exemple, vous pouvez créer un journal dont le type de données d'enregistrement est un cluster d'une chaîne et d'un nombre. Chaque enregistrement du journal sera un cluster d'une chaîne et d'un nombre. Cependant, le premier enregistrement pourrait être ("abc" , 1), et le deuxième ("xyz" , 7).

L'utilisation de fichiers journaux ne nécessite pas beaucoup de manipulations, ce qui rend l'écriture et la lecture beaucoup plus rapides. Elle simplifie aussi la récupération des données puisque vous pouvez relire les blocs de données initiaux en tant qu'enregistrement sans avoir à lire tous les enregistrements qui les précèdent. L'accès aléatoire est rapide et facile avec les fichiers journaux, puisque tout ce dont vous avez besoin pour accéder à l'enregistrement est le numéro de cet enregistrement. LabVIEW assigne de manière séquentielle un numéro d'enregistrement à chaque enregistrement lorsqu'il crée le fichier journal.

Vous pouvez accéder à des fichiers journaux à partir de la face-avant et à partir du diagramme. Reportez-vous à la section [Enregistrement des données de la face-avant](#) dans ce chapitre pour obtenir plus de détails sur l'accès à des fichiers journaux à partir de la face-avant.

LabVIEW écrit un enregistrement dans un fichier journal à chaque exécution du VI qui lui est associé. Vous ne pouvez pas écraser un enregistrement écrit par LabVIEW dans un fichier journal. Lorsque vous lisez un fichier journal, vous pouvez lire un seul enregistrement ou plusieurs enregistrements en même temps.

Reportez-vous à `exemples\file\data\log.llb` si vous souhaitez consulter des exemples de lecture et d'écriture de fichiers journaux.

Utilisation de VIs d'E/S sur fichiers de haut niveau

Utilisez les VIs d'E/S sur fichiers de haut niveau situés dans la ligne du haut de la palette **Fonctions»E/S sur fichiers** pour effectuer des opérations d'E/S courantes, comme la lecture ou l'écriture des types de données suivants :

- Caractères vers ou à partir de fichiers texte
- Lignes à partir de fichiers texte
- Tableaux 1D ou 2D de numériques simple précision vers ou à partir de fichiers texte de type tableur
- Tableaux 1D ou 2D de numériques simple précision ou d'entiers 16 bits signés vers ou à partir de fichiers binaires

Vous pouvez gagner du temps et rendre la programmation plus facile en utilisant les VIs de haut niveau pour écrire et lire dans des fichiers. Les VIs de haut niveau ouvrent et ferment le fichier et effectuent aussi des opérations de lecture et d'écriture. Évitez de placer les VIs de haut-niveau dans des boucles car les VIs effectuent des opérations d'ouverture et de fermeture à chaque fois qu'elles s'exécutent. Reportez-vous à la section [Enregistrement sur disque en continu](#) pour obtenir plus d'informations sur la façon de garder les fichiers ouverts lorsque vous effectuez des opérations multiples.

Les VIs de haut niveau attendent une entrée de chemin de fichier. Si vous ne câblez pas de chemin de fichier, une boîte de dialogue apparaît et vous demande de spécifier dans quel fichier vous voulez lire ou écrire. Si une erreur survient, les VIs de haut niveau affichent une boîte de dialogue qui décrit cette erreur. Vous avez le choix entre interrompre l'exécution ou la poursuivre.

La figure 13-1 montre comment utiliser le VI de haut niveau Écrire dans un fichier tableau pour envoyer des nombres vers un fichier tableau Excel. Lorsque vous exécutez ce VI, LabVIEW vous invite à écrire les données dans un fichier existant ou à créer un nouveau fichier.

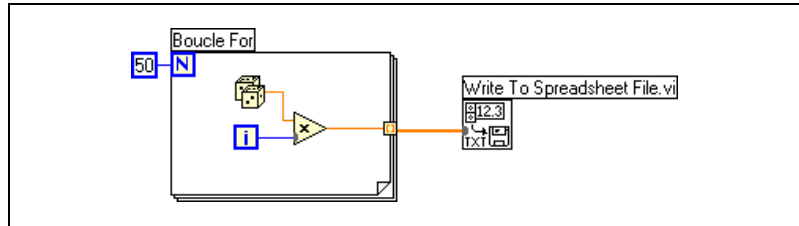


Figure 13-1. Utilisation d'un VI de haut niveau pour écrire dans un fichier tableau

Utilisez les VIs de fichiers binaires, situés dans la palette **Fonctions»E/S sur fichiers»VIs de fichiers binaires** pour lire et écrire dans des fichiers au format binaire. Les données peuvent être des entiers ou des flottants simple précision.

Utilisation de VIs et de fonctions d'E/S sur fichiers de bas niveau et avancés

Utilisez les VIs et fonctions d'E/S sur fichiers de bas niveau, situés dans la rangée du milieu de la palette **Fonctions»E/S sur fichiers** et les fonctions avancées d'E/S sur fichiers, situées dans la palette **Fonctions»E/S sur fichiers»Fonctions de fichiers avancées**, pour contrôler une par une les opérations d'E/S sur fichiers.

Utilisez les principales fonctions de bas niveau pour créer ou ouvrir un fichier, écrire des données ou lire des données dans le fichier et pour fermer le fichier. Utilisez les autres fonctions de bas niveau pour effectuer les tâches suivantes :

- Créer des répertoires.
- Déplacer, copier ou supprimer des fichiers.
- Lister le contenu d'un répertoire.
- Modifier les caractéristiques d'un fichier.
- Manipuler des chemins.



Un chemin, comme illustré à gauche, est un type de données LabVIEW qui identifie l'emplacement d'un fichier sur le disque. Le chemin décrit le disque qui contient le fichier, les répertoires situés entre le niveau principal

du système de fichiers et le fichier, et le nom du fichier. Entrez ou affichez un chemin au moyen de la commande ou de l'indicateur de chemin, en utilisant la syntaxe standard de la plate-forme. Reportez-vous à la section *Commandes et indicateurs de type chemin* du chapitre 4, *Construction de la face-avant*, pour plus d'informations sur les commandes et indicateurs de chemin.

La figure 13-2 montre comment utiliser les VIs et les fonctions de bas niveau pour envoyer des nombres vers un fichier tableur Microsoft Excel. Lorsque vous exécutez ce VI, le VI Ouvrir/Créer/Remplacer un fichier ouvre le fichier `numbers.xls`. La fonction Écrire dans un fichier écrit la chaîne de nombres dans le fichier. La fonction Fermer un fichier ferme le fichier. Si vous ne fermez pas le fichier, celui-ci reste dans la mémoire et n'est pas accessible à partir d'autres applications, ni par d'autres utilisateurs.

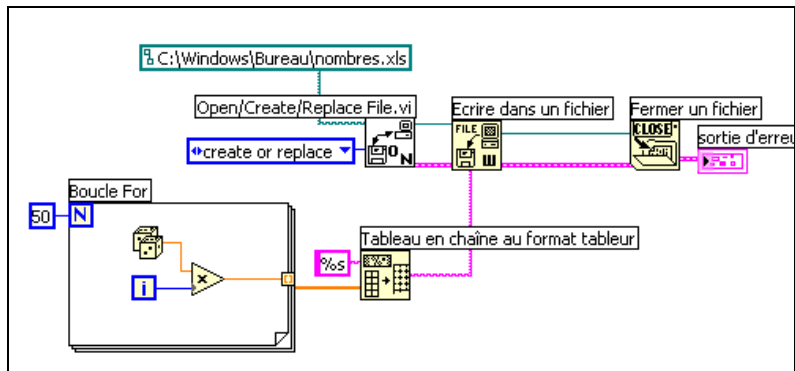


Figure 13-2. Utilisation d'un VI de bas niveau pour écrire dans un fichier tableur

Comparez le VI de la figure 13-2 au VI de la figure 13-1, qui exécute la même tâche. Dans la figure 13-2, vous devez utiliser la fonction Tableau en chaîne au format tableur pour convertir un tableau de nombres au format chaîne. Le VI Écrire dans un fichier tableur de la figure 13-1 ouvre le fichier, convertit le tableau de nombres en chaîne et ferme le fichier.

Reportez-vous au VI Write Datalog File dans `examples\file\datalog.llb` pour visualiser un exemple d'utilisation des VIs et des fonctions d'E/S sur fichiers de bas niveau.

Enregistrement sur disque en continu

Vous pouvez aussi utiliser les VIs et les fonctions d'E/S sur fichiers de bas niveau pour enregistrer sur le disque en continu, ce qui permet d'économiser les ressources mémoire. L'enregistrement sur disque en continu est une technique qui permet de garder des fichiers ouverts pendant

que vous effectuez par exemple, plusieurs opérations d'écriture à l'intérieur d'une boucle. Ces opérations d'écriture de haut niveau sont faciles à utiliser mais elles obligent l'utilisateur à ouvrir et à fermer le fichier à chacune de leurs exécutions. Vos VIs seront plus performants si vous évitez d'ouvrir et de fermer les mêmes fichiers trop fréquemment.

L'enregistrement sur disque en continu réduit le nombre d'interactions entre la fonction et le système d'exploitation au moment de l'ouverture et de la fermeture de fichiers. Pour créer une opération normale d'enregistrement sur disque en continu, placez le VI Ouvrir/Créer/Remplacer un fichier avant la boucle, et la fonction Fermer un fichier après la boucle. Vous pouvez alors écrire en continu dans un fichier sans avoir à l'ouvrir et à le fermer.

L'enregistrement en continu est idéal pour les opérations d'acquisition de données très longues dans lesquelles la vitesse est un point crucial. Vous pouvez écrire des données en continu dans un fichier pendant que l'acquisition se poursuit. Pour obtenir les meilleurs résultats possibles, évitez d'utiliser d'autres fonctions, telles que les fonctions d'analyse, avant la fin de l'acquisition.

Création de fichiers texte et de fichiers tableur

Pour écrire des données dans un fichier texte ou dans un fichier de type tableur, vous devez convertir vos données en chaîne. Pour écrire des données dans un fichier de type tableur, vous devez convertir la chaîne en chaîne de type tableur, qui est une chaîne comportant des séparateurs, tels que des tabulateurs. Reportez-vous à la section *Formatage de chaînes* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir plus de détails sur le formatage des chaînes.

L'écriture de texte dans des fichiers texte ne nécessite pas de formatage parce que la plupart des applications de traitement de texte peuvent lire du texte non formaté. Pour écrire une chaîne texte dans un fichier texte, utilisez le VI Écrire des caractères dans un fichier, qui ouvre et ferme le fichier automatiquement.

Utilisez le VI Écrire dans un fichier tableur ou la fonction Tableau en chaîne au format tableur pour convertir en chaîne de type tableur un ensemble de nombres à partir d'un graphe, un graphe déroulant ou une acquisition. Reportez-vous aux sections *Utilisation de VIs d'E/S sur fichiers de haut niveau* et *Utilisation de VIs et de fonctions d'E/S sur fichiers de bas niveau et avancés* dans ce chapitre, pour obtenir plus de détails sur l'utilisation de ces VIs et de ces fonctions.

La lecture de texte à partir d'une application de traitement de texte peut engendrer des erreurs, parce que les applications de traitement de texte formatent du texte en utilisant des polices de caractères, des couleurs, des styles et des tailles que les VIs de fichiers tableur ne peuvent pas traiter.

Si vous voulez écrire des nombres et du texte dans une application de type tableur ou traitement de texte, utilisez les fonctions Chaîne, situées dans la palette **Fonctions»Chaîne** et les fonctions Tableau, situées dans la palette **Fonctions»Tableau** pour formater les données et concaténer les chaînes. Écrivez ensuite les données dans un fichier. Reportez-vous au chapitre 9, [Groupement des données au moyen de chaînes, de tableaux et de clusters](#), afin d'obtenir plus de détails sur l'utilisation de ces fonctions pour formater et concaténer des données.

Formatage et écriture de données dans des fichiers

Utilisez la fonction Formater dans un fichier pour formater des données chaîne, numériques, booléennes et de chemin en texte et pour écrire le texte formaté dans un fichier. Vous pouvez souvent utiliser cette fonction plutôt que de formater la chaîne au moyen de la fonction Formater en chaîne, puis d'écrire la chaîne résultante au moyen du VI Écrire des caractères dans un fichier ou de la fonction Écrire dans un fichier.

Vous pouvez ajouter ou écraser des données existantes dans un fichier avec la fonction Formater dans un fichier. Utilisez la fonction Rechercher pour placer un repère de fichier dans un refnum. Après que la fonction Formater dans un fichier a écrit la chaîne résultante, le repère de fichier est placé à la fin du texte.

Reportez-vous à la section [Formatage de chaînes](#) du chapitre 9, [Groupement des données au moyen de chaînes, de tableaux et de clusters](#), pour obtenir plus de détails sur le formatage de chaînes.

Balayage de données à partir de fichiers

Utilisez la fonction Balayer à partir d'un fichier afin d'y rechercher les valeurs des chaînes, des nombres, des chemins et les valeurs booléennes, et pour les convertir ensuite en type de données. Vous pouvez souvent utiliser cette fonction plutôt que de lire les données à partir d'un fichier au moyen de la fonction Lire un fichier ou du VI Lire des caractères dans un fichier et de lire ensuite la chaîne résultante au moyen de la fonction Balayer une chaîne.

Utilisez la fonction Balayer à partir d'un fichier pour lire le texte formaté dans le fichier. Utilisez la fonction Rechercher pour placer le repère à l'endroit où le balayage commence.

Création de fichiers binaires

Pour créer un fichier binaire, vous devez acquérir un ensemble de nombres et les écrire dans un fichier. Vous n'avez pas à formater les nombres, sauf si vous voulez utiliser les VIs d'E/S sur fichiers binaires de haut niveau situés dans la palette **Fonctions»E/S sur fichiers»VIs de fichiers binaires**.

Utilisez les VIs Écrire dans un fichier I16 et Écrire dans un fichier SGL pour enregistrer des tableaux 1D et 2D de nombres dans un fichier. Vous devez tout d'abord convertir les nombres en entiers 16 bits ou en flottants simple précision. Utilisez les VIs Lire dans un fichier I16 et Lire dans un fichier SGL pour lire les fichiers que vous avez créés.

Pour écrire des nombres de types de données différents, tels que des flottants double précision ou des entiers 32 bits non signés, utilisez les fonctions de fichiers de bas niveau et avancées, situées dans la palette **Fonctions»E/S sur fichiers»Fonctions de fichiers avancées**. Lorsque vous lisez le fichier, utilisez la fonction Lire un fichier et spécifiez le type de données du nombre.

Reportez-vous aux VIs Read Binary File et Write Binary File dans `examples\file\smpfile.llb` si vous souhaitez consulter des exemples de lecture et d'écriture de flottants simples dans un fichier binaire.

Création de fichiers journaux

Vous pouvez créer et lire des fichiers journaux en activant le journal de la face-avant ou en utilisant les fonctions de fichier de bas niveau ou avancées situées dans la palette **Fonctions»E/S sur fichiers»Fonctions de fichiers avancées** pour acquérir des données et les écrire dans un fichier.

Reportez-vous à la section [Enregistrement des données de la face-avant](#) dans ce chapitre pour obtenir plus de détails sur la création de fichiers journaux et leur accès à partir de la face-avant.

Vous n'avez pas à formater les données d'un fichier journal. Cependant, lorsque vous écrivez ou que vous lisez des fichiers journaux, vous devez spécifier le type des données. Par exemple, si vous acquérez une lecture de température mentionnant l'heure et la date auxquelles la température a été

enregistrée, vous écrivez les données dans un fichier journal et vous spécifiez les données comme cluster d'un nombre et de deux chaînes. Reportez-vous au VI Simple Temp Datalogger dans `examples\file\datalog.llb` si vous voulez consulter un exemple d'écriture de données dans un fichier journal.

Si vous lisez un fichier incluant une lecture de température mentionnant l'heure et la date auxquelles la température a été enregistrée, spécifiez que vous voulez lire un cluster d'un nombre et de deux chaînes. Reportez-vous au VI Simple Temp Datalog Reader dans `examples\file\datalog.llb` si vous voulez consulter un exemple de lecture de fichier journal.

Écrire des données waveform dans un fichier

Utilisez les VIs *Écrire des waveforms dans un fichier* et *Exporter des waveforms dans un fichier tableur*, situés dans la palette **Fonctions»Waveform»E/S sur fichiers de waveform** pour envoyer des données waveform dans des fichiers. Vous pouvez écrire des données waveform dans des fichiers tableur, des fichiers texte ou dans des fichiers journaux.

Si vous avez l'intention d'utiliser les données waveform que vous avez créées dans un seul VI, enregistrez-les en tant que fichier journal (.log). Reportez-vous à la section *Quand utiliser des fichiers journaux* dans ce chapitre pour obtenir plus de détails sur les fichiers journaux.

Le VI de la figure 13-3 acquiert des signaux multiples, les affiche, sous forme de données waveform, dans un graphe et les écrit dans un fichier tableur de Microsoft Excel.

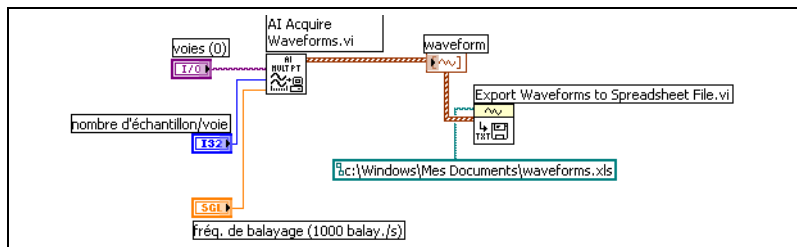


Figure 13-3. Écriture de multiples waveforms dans un fichier tableur

Lecture de données waveform dans des fichiers

Utilisez le VI Lire une waveform à partir d'un fichier situé dans la palette **Fonctions»Waveform»E/S sur fichiers de waveform** pour lire des données waveform à partir d'un fichier. Après avoir lu une waveform simple, vous pouvez ajouter ou éditer des composantes de données waveform au moyen de la fonction Construire une waveform située dans la palette **Fonctions»Waveform**, ou vous pouvez extraire des composantes de données waveform au moyen de la fonction Obtenir un attribut de waveform, située dans la palette **Fonctions»Waveform**.

Le VI de la figure 13-4 lit des données waveform dans un fichier, édite la composante **dt** et trace les données waveform éditées dans un graphe.

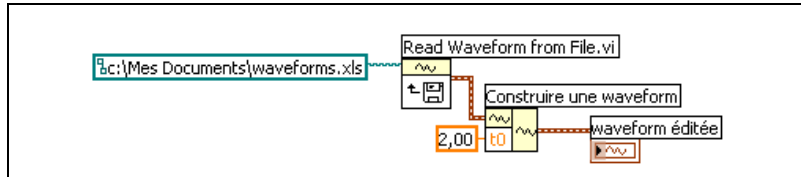


Figure 13-4. Lecture d'une waveform dans un fichier

Le VI Lire une waveform à partir d'un fichier lit aussi des waveforms multiples dans un fichier. Le VI renvoie un tableau de données de type waveform, que vous pouvez afficher dans un graphe à courbes multiples. Si vous voulez accéder à une waveform unique dans un fichier, vous devez indexer le tableau de types de données waveform, comme l'illustre la figure 13-5. Reportez-vous à la section *Tableaux* du chapitre 9, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur l'indexation de tableaux. Le VI accède à un fichier qui comporte des waveforms multiples. La fonction Indexer un tableau lit la première et la troisième waveforms dans le fichier et les trace dans deux graphes séparés. Reportez-vous au chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement* pour obtenir plus de détails sur les graphes.

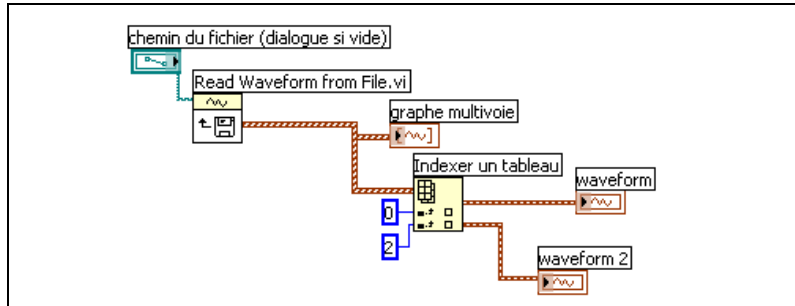


Figure 13-5. Lecture de waveforms multiples dans un fichier

Paramètres dupliqués

De nombreux VIs et fonctions d'E/S sur fichiers comportent des paramètres dupliqués, habituellement un refnum ou un chemin, qui renvoient la même valeur que le paramètre d'entrée qui leur correspond. Utilisez ces paramètres pour contrôler l'ordre d'exécution des fonctions. En câblant la sortie dupliquée du premier nœud que vous voulez exécuter à l'entrée correspondante du nœud que vous voulez exécuter ensuite, vous créez une dépendance artificielle de données. Sans ces paramètres dupliqués, il vous faut utiliser des structures Séquence pour vous assurer que les opérations d'E/S sur fichiers se déroulent dans l'ordre que vous souhaitez.

Reportez-vous à la section [Dépendance des données et dépendance des données artificielle](#) du chapitre 5, [Construction du diagramme](#), pour obtenir plus d'informations sur la dépendance artificielle de données.

Création de fichiers de configuration

Utilisez les VIs de fichiers de configuration situés dans la palette **Fonctions»E/S sur fichiers»VIs de fichiers de configuration** pour lire et créer des fichiers de paramètres de configuration Windows standard (.ini) et pour écrire des données propres à la plate-forme, tels que des chemins, au format indépendant de la plate-forme de manière à pouvoir utiliser les fichiers générés par ces VIs sur plusieurs plates-formes.

Reportez-vous à `examples\file\config.llb` si vous voulez consulter des exemples d'utilisation de VIs de fichiers de configuration.



Remarque L'extension standard des fichiers de paramètres de configuration Windows est .ini, mais les VIs de fichiers de configuration fonctionnent avec n'importe quelle autre extension de fichier, du moment que le contenu du fichier est au bon format. Reportez-vous

à la section *Format de fichier de paramètres de configuration Windows* dans ce chapitre pour obtenir plus de détails sur la configuration du contenu.

Utilisation de fichiers de paramètres de configuration

Un fichier de paramètres de configuration standard Windows est un format particulier pour le stockage des données dans un fichier texte. Vous pouvez facilement accéder par programmation aux données à l'intérieur d'un fichier `.ini` parce qu'il suit un format particulier.

Par exemple, prenez un fichier de paramètres de configuration dont le contenu est le suivant :

```
[Donnée]
Valeur = 7,2
```

Vous pouvez utiliser les VIs de fichiers de configuration pour lire ces données, comme l'illustre la figure 13-6. Ce VI utilise le VI Lire une clé pour lire la **clé** appelée `Valeur` dans la **section** appelée `Donnée`. Ce VI fonctionne sans tenir compte des modifications apportées au fichier, pourvu que le fichier reste au format de fichier de paramètres de configuration Windows.

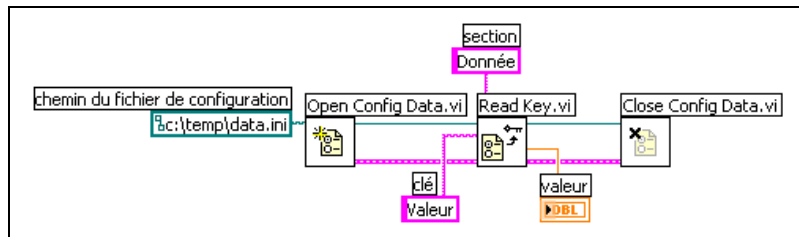


Figure 13-6. Lecture des données à partir d'un fichier `.ini`

Format de fichier de paramètres de configuration Windows

Les fichiers de paramètres de configuration Windows sont des fichiers texte divisés en sections nommées. Des parenthèses entourent chaque nom de section. Chaque section porte un nom différent. Les sections contiennent des paires clé/valeur séparées par le signe égal (=). Les noms situés à l'intérieur de chaque section sont tous différents. Le nom de clé représente une préférence de configuration, et le nom de valeur représente le paramètre de cette préférence. L'exemple suivant illustre l'organisation du fichier :

```
[Section 1]
clé1=valeur
```

```

clé2=valeur
[Section 2]
clé1=valeur
clé2=valeur

```

Utilisez les types de données suivants pour la partie qui concerne la valeur du paramètre **clé** dans les VIs de fichiers de configuration :

- Chaîne
- Chemin
- Booléen
- Flottant double précision 64 bits
- Entier 32 bits signé
- Entier 32 bits non signé

Les VIs de fichiers de configuration peuvent lire et écrire des données chaîne brute ou échappée. Les VIs lisent et écrivent les données brutes octet par octet, sans convertir les données en code ASCII. Dans des chaînes converties ou échappées, LabVIEW stocke tout texte non affichable dans le fichier de paramètres de configuration avec les codes hexadécimaux équivalents, comme `\0D` pour un retour chariot. De plus, LabVIEW stocke les barres obliques inverses dans le fichier de paramètres de configuration en double, c'est-à-dire `\\` pour `\`. Définissez les entrées **lire la chaîne brute ?** ou **écrire la chaîne brute ?** des VIs de fichiers de configuration sur VRAI pour les données brutes et sur FAUX pour les données de chaîne échappée.

Lorsque les VIs écrivent dans un fichier de configuration, ils mettent entre guillemets toutes les données chaîne et les données de chemin qui contiennent une espace. Si une chaîne comporte des guillemets, LabVIEW les stocke comme `\"`. Si vous lisez et/ou écrivez dans des fichiers de configuration qui utilisent un éditeur de texte, vous pouvez remarquer que LabVIEW a remplacé les guillemets par des `\"`.

LabVIEW stocke les données de chemin en format indépendant de la plate-forme, le format UNIX standard pour chemins, dans des fichiers `.ini`. Les VIs interprètent le chemin absolu `/c/temp/data.dat` stocké dans le fichier de paramètres de configuration comme suit :

- **(Windows)** `c:\temp\data.dat`
- **(Macintosh)** `c:temp:data.dat`
- **(UNIX)** `/c/temp/data.dat`

Les VIs interprètent le chemin relatif `temp\data.dat` comme suit :

- **(Windows)** `temp\data.dat`
- **(Macintosh)** `:temp:data.dat`
- **(UNIX)** `temp/data.dat`

Enregistrement des données de la face-avant

Utilisez le journal de la face-avant pour enregistrer des données à utiliser dans d'autres VIs et dans des rapports. Par exemple, vous pouvez enregistrer des données à partir d'un graphe et les utiliser dans un graphe de VI différent.

Chaque fois qu'un VI s'exécute, le journal de la face-avant enregistre les données de la face-avant dans un fichier journal distinct au format texte délimité. Vous pouvez extraire les données de plusieurs manières :

- Utilisez le VI à partir duquel vous avez enregistré les données pour extraire les données de manière interactive.
- Utilisez le VI comme un sous-VI pour extraire les données par programmation.
- Utilisez les VIs et fonctions d'E/S sur fichiers pour extraire les données.

Chaque VI entretient une liaison de fichier journal qui enregistre l'endroit du fichier journal où LabVIEW conserve les données de la face-avant enregistrées. La liaison de fichier journal représente l'association entre un VI et le fichier journal dans lequel vous enregistrez les données du VI.

Un fichier journal contient des enregistrements qui incluent un horodatage et les données enregistrées à chaque exécution du VI. Lorsque vous accédez à un fichier journal, vous sélectionnez un enregistrement en exécutant le VI en mode d'extraction et en vous servant des commandes de la face-avant pour visualiser les données. Lorsque vous exécutez le VI en mode d'extraction, une commande numérique qui vous permet de naviguer dans les enregistrements s'affiche en haut de la face-avant. Reportez-vous à la figure 13-7 pour obtenir un exemple de cette commande numérique.

Journaux de la face-avant automatiques et interactifs

Sélectionnez **Exécution»Enregistrer à la fin de l'exécution** pour activer l'enregistrement automatique. La première fois que vous enregistrez des données de la face-avant pour un VI, LabVIEW vous invite à nommer le fichier journal. Chaque fois que vous exécutez le VI, LabVIEW enregistre

les données et ajoute un nouvel enregistrement au fichier journal. Vous ne pouvez pas écraser un enregistrement écrit par LabVIEW dans un fichier journal.

Pour enregistrer vos données de manière interactive, sélectionnez **Exécution»Fichier journal»Enregistrer....** LabVIEW ajoute immédiatement les données au fichier journal. Lorsque vous enregistrez vos données de manière interactive, vous pouvez choisir le moment où les données doivent être enregistrées. La fonction d'enregistrement automatique des données enregistre les données à chaque exécution du VI.



Remarque Un graphe déroulant n'enregistre qu'un point de données dans le journal de la face-avant. Si vous câblez un tableau à l'indicateur du graphe déroulant, le fichier journal va contenir un sous-ensemble du tableau que le graphe déroulant affiche.

Visualisation interactive des données enregistrées de la face-avant

Après avoir enregistré des données, vous pouvez les visualiser de manière interactive en sélectionnant **Exécution»Fichier journal»Retrouver....** La barre d'outils de récupération des données s'affiche, comme l'illustre la figure 13-7.

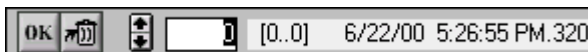


Figure 13-7. Barre d'outils de récupération de données

Le chiffre en surbrillance indique l'enregistrement de données que vous visualisez actuellement. Les chiffres entre crochets indiquent la gamme d'enregistrements que vous avez effectués pour le VI actif. Vous effectuez un enregistrement chaque fois que vous exécutez le VI. La date et l'heure indiquent quand l'enregistrement sélectionné a été effectué. Visualisez l'enregistrement suivant ou précédent en cliquant sur les flèches d'incrémentation et de décrémentation. Vous pouvez aussi utiliser les flèches haut et bas du clavier.

L'apparence de la face-avant change en fonction de l'enregistrement que vous sélectionnez dans la barre d'outils. Par exemple, lorsque vous cliquez sur la flèche d'incrémentation et que vous passez à un autre enregistrement, les commandes et l'indicateur affichent les données relatives à cet enregistrement telles qu'elles ont été enregistrées. Cliquez sur le bouton **OK** pour quitter le mode de récupération et revenir au VI dont vous visualisez actuellement le fichier journal.

Effacer un enregistrement

Lorsque vous êtes en mode de récupération, vous pouvez effacer des enregistrements. Pour effacer un enregistrement en mode de récupération, visualisez cet enregistrement et sélectionnez-le comme enregistrement à effacer en cliquant sur le bouton **Corbeille**. Si vous cliquez à nouveau sur le bouton **Corbeille**, vous annulez la sélection.

Sélectionnez **Exécution»Fichier journal»Vider les données...** lorsque vous êtes en mode de récupération pour effacer tous les enregistrements que vous avez cochés.

Si vous ne sélectionnez pas **Exécution»Fichier journal»Vider les données...** avant de cliquer sur le bouton **OK**, LabVIEW vous invite à effacer les enregistrements que vous avez cochés.

Effacer la liaison vers le fichier journal

La liaison vers le fichier journal sert à associer un VI au fichier journal à utiliser lorsque vous enregistrez ou récupérez les données de la face-avant. Deux fichiers journaux ou plus peuvent être associés à un VI. Cette fonction est utile pour tester ou comparer les données du VI. Par exemple, vous pouvez comparer les données enregistrées à la première exécution du VI aux données enregistrées à la deuxième exécution. Pour associer plusieurs fichiers journaux à un VI, vous devez effacer la liaison vers le fichier journal en sélectionnant **Exécution»Fichier journal»Effacer la liaison vers le fichier journal**. Lors de l'exécution suivante du VI, LabVIEW vous invite, lorsque vous spécifiez un fichier journal, à activer la commande d'enregistrement automatique ou à choisir à quel moment vous voulez enregistrer les données de manière interactive.

Changer la liaison vers le fichier journal

Changez la liaison vers le fichier journal, pour enregistrer ou récupérer les données de la face-avant dans un fichier journal différent, en sélectionnant **Exécution»Fichier journal»Changer la liaison vers le fichier journal...** LabVIEW vous propose de sélectionner un fichier journal différent ou d'en créer un nouveau. Vous pouvez changer la liaison vers le fichier journal lorsque vous voulez récupérer différentes données dans un VI ou ajouter les données d'un VI à un autre fichier journal.

Récupération des données de la face-avant par programmation

Vous pouvez aussi récupérer des données enregistrées en vous servant d'un sous-VI ou des VIs et des fonctions d'E/S sur fichiers.

Récupération des données de la face-avant au moyen d'un sous-VI

Lorsque vous cliquez sur un sous-VI avec le bouton droit de la souris et que vous sélectionnez **Autoriser l'accès à la base de données** dans le menu local, une boîte jaune apparaît autour du sous-VI, comme l'illustre la figure 13-8.

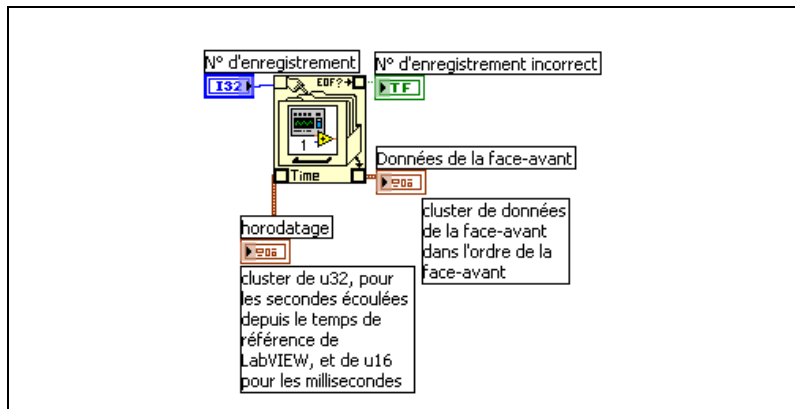


Figure 13-8. Récupération des données enregistrées

La boîte jaune, qui ressemble à un classeur, est munie de terminaux d'accès aux données à partir du fichier journal. Lorsque vous autorisez le sous-VI à accéder à la base de données, les entrées et les sorties du sous-VI se comportent comme des sorties et renvoient leurs données enregistrées. Le **numéro d'enregistrement** indique l'enregistrement à récupérer, le **numéro d'enregistrement incorrect** indique si le numéro d'enregistrement existe, l'**horodatage** correspond à l'heure à laquelle l'enregistrement a été créé et **données de la face-avant** est un cluster d'objets de la face-avant. Vous pouvez accéder aux données d'un objet de la face-avant en câblant le cluster **données de la face-avant** à la fonction Désassembler.

Vous pouvez aussi récupérer des valeurs pour des entrées et des sorties particulières en vous câblant directement au terminal correspondant sur le sous-VI, comme l'illustre la figure 13-9.

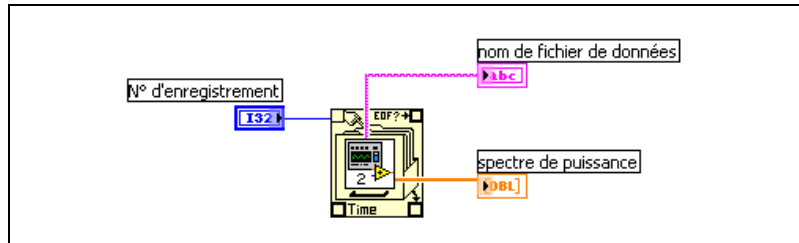


Figure 13-9. Récupération des données enregistrées par l'intermédiaire de terminaux de sous-VIs

Si vous exécutez le VI, le sous-VI ne s'exécute pas. Il renvoie les données enregistrées à partir de sa face-avant dans la face-avant du VI, sous la forme d'un cluster.

Spécifier les enregistrements

Le sous-VI comporte n enregistrements et vous pouvez câbler n'importe quel nombre de $-n$ à $n - 1$ au terminal **numéro d'enregistrement** du sous-VI. Vous pouvez accéder aux enregistrements par rapport au premier enregistrement effectué au moyen de numéros d'enregistrement non négatifs. 0 représente le premier enregistrement, -1 représente le deuxième enregistrement et ainsi de suite jusqu'à $n - 1$, qui représente le dernier enregistrement.

Vous pouvez accéder aux enregistrements par rapport au dernier enregistrement effectué au moyen de numéros d'enregistrement négatifs. -1 représente le dernier enregistrement, -2 représente l'avant-dernier enregistrement, et ainsi de suite jusqu'à $-n$, qui représente le premier enregistrement. Si vous câblez un nombre hors de la gamme $-n$ à $n - 1$ au terminal du **numéro d'enregistrement**, la sortie du **numéro d'enregistrement incorrect** affiche VRAI et le sous-VI ne récupère pas de données.

Récupération des données de la face-avant au moyen des fonctions d'E/S sur fichiers

Vous pouvez aussi récupérer des données que vous avez enregistrées à partir de la face-avant, au moyen des VIs et des fonctions d'E/S sur fichiers, comme la fonction Lire un fichier. Le type de données de chaque enregistrement dans le fichier journal de la face-avant crée deux clusters. L'un des clusters contient un horodatage et l'autre cluster contient les données de la face-avant. Le cluster d'horodatage inclut un entier 32 bits non signé qui représente les secondes et un entier 16 bits non signé qui représente les millisecondes écoulées depuis l'heure de référence de LabVIEW, c'est-à-dire minuit, le 1er janvier 1904.

Vous accédez aux enregistrements des fichiers journaux de la face-avant au moyen des mêmes fonctions d'E/S sur fichiers que vous utilisez pour accéder aux fichier journaux que vous avez créés par programmation. Entrez le **type d'enregistrement de données** comme entrée de type dans la fonction Ouvrir un fichier, comme l'illustre la figure 13-10.

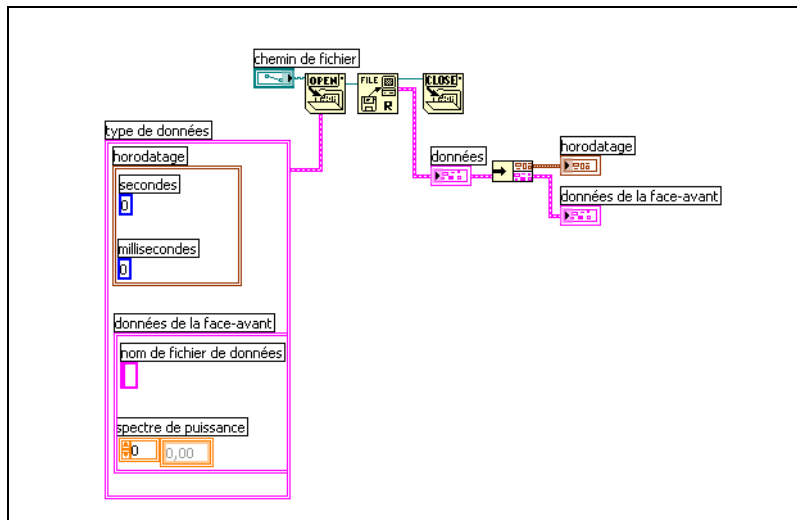


Figure 13-10. Récupération des données enregistrées à l'aide de la fonction Ouvrir un fichier

Documenter et imprimer des VIs

Vous pouvez utiliser LabVIEW pour documenter et imprimer les VIs.

La section *Documentation des VIs* de ce chapitre décrit comment enregistrer des informations sur le diagramme et/ou la face-avant à n'importe quel stade du développement sous forme de documentation imprimée sur les VIs.

La section *Impression des VIs* de ce chapitre passe en revue les options d'impression pour les VIs. Certaines options conviennent mieux à l'impression des informations sur les VIs et d'autres conviennent mieux aux rapports sur les données et les résultats que les VIs génèrent. La méthode d'impression à utiliser dépend de plusieurs facteurs, comme le choix entre l'impression manuelle ou l'impression par programmation, le nombre d'options dont vous avez besoin pour le format du rapport, l'utilisation ou non de la fonctionnalité dans les applications autonomes que vous construisez et le choix des plates-formes sur lesquelles vous exécutez les VIs.

La section *Impression de rapports* de ce chapitre décrit comment imprimer des rapports dans LabVIEW.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour de plus amples informations sur la documentation et l'impression des VIs.

Documentation des VIs

Vous pouvez utiliser LabVIEW pour suivre un développement, documenter un VI terminé et créer des instructions pour les utilisateurs des VIs. Vous pouvez visualiser la documentation à l'intérieur de LabVIEW, l'imprimer et l'enregistrer dans des fichiers au format HTML ou RTF.



Remarque Vous ne pouvez pas imprimer de documentation de VIs, quel que soit le format, dans les applications intégrées.

Pour utiliser une documentation imprimée le plus efficacement possible, créez des descriptions de VIs et d'objets et activez l'historique des révisions des VIs.

Création de descriptions de VIs et d'objets

Créez des descriptions pour les VIs et leurs objets, comme les commandes et indicateurs, pour expliquer dans quel but le VI et l'objet ont été créés et pour fournir des instructions sur l'utilisation du VI ou de l'objet aux utilisateurs. Vous pouvez visualiser les descriptions dans LabVIEW ou les imprimer.

Créez, éditez et visualisez les descriptions des VIs en sélectionnant **Fichier»Propriétés du VI** et en sélectionnant **Documentation** dans le menu déroulant **Catégorie**. Créez, éditez et visualisez les descriptions d'objets et de sous-VIs en cliquant sur l'objet avec le bouton droit de la souris et en sélectionnant **Description et info-bulle** dans le menu local. Les info-bulles sont de brèves descriptions qui apparaissent lorsque vous positionnez le curseur sur un objet. Si vous n'entrez pas d'info-bulle dans la boîte de dialogue **Description et info-bulle**, aucune info-bulle ne s'affiche. La description du VI ou de l'objet s'affiche aussi dans la fenêtre **Aide contextuelle** lorsque vous déplacez le curseur sur l'icône du VI ou sur l'objet. Sélectionnez **Aide»Aide contextuelle** pour afficher la fenêtre d'**Aide contextuelle**. La description du VI ou de l'objet apparaît également dans toute documentation de VI que vous générez.

Activation de l'historique des révisions des VIs

Utilisez la fenêtre **Historique** dans chaque VI pour afficher l'historique du développement du VI, les numéros de révision y compris. Enregistrez et repérez les modifications apportées au VI dans la fenêtre **Historique** au fur et à mesure que vous les effectuez. Sélectionnez **Outils»Historique des révisions du VI** pour ouvrir la fenêtre **Historique**. Vous pouvez également imprimer l'historique des révisions. Reportez-vous à la section [Impression de la documentation](#) de ce chapitre pour obtenir plus de détails sur l'impression de l'historique des révisions du VI.

Numéros de révision

Le numéro de révision permet de repérer facilement les modifications apportées à un VI. Le numéro de révision commence à zéro et augmente selon une progression régulière chaque fois que vous enregistrez le VI. Le numéro de révision actuel stocké sur le disque s'affiche dans la fenêtre **Historique**. Pour afficher le numéro de révision actuel dans la barre de titre du VI, sélectionnez **Outils»Options...**, sélectionnez **Historique des**

révisions dans le menu déroulant du haut et cochez la case **Visualiser le numéro de révision dans la barre de titre**.

Le numéro que LabVIEW affiche dans la barre de titre de la fenêtre **Historique** et dans la barre de titre du VI correspond au numéro de révision suivant, c'est-à-dire au numéro de révision actuel plus un. Lorsque vous ajoutez un commentaire à l'historique, l'en-tête du commentaire inclut le numéro de révision suivant. Le numéro de révision n'augmente pas lorsque vous enregistrez un VI si vous modifiez uniquement la fenêtre **Historique**.

Les numéros de révision sont indépendants des commentaires de la fenêtre **Historique**. Les chiffres manquants des numéros de révision entre les commentaires indiquent que vous avez enregistré le VI sans commentaire.

Comme l'historique est exclusivement un outil de développement, LabVIEW le supprime automatiquement lorsque vous supprimez le diagramme d'un VI. Reportez-vous à la section *Distribution des VIs* du chapitre 7, *Création de VIs et de sous-VIs*, pour de plus amples informations sur la suppression du diagramme. La fenêtre **Historique** n'est pas accessible dans la version d'exécution d'un VI. La page **Général** de la boîte de dialogue **Propriétés du VI** affiche le numéro de révision, même pour les VIs sans diagramme. Cliquez sur le bouton de **Remise à zéro** dans la fenêtre **Historique** pour remettre le numéro de révision à zéro.

Impression de la documentation

Sélectionnez **Fichier»Imprimer...** pour imprimer la documentation d'un VI ou l'enregistrer dans un fichier HTML ou RTF. Vous pouvez sélectionner un format existant ou créer votre propre format pour la documentation. La documentation que vous créez peut inclure les éléments suivants :

- Cadre connecteur et icône
- Face-avant et diagramme
- Commandes, indicateurs, et types de données
- Hiérarchie du VI
- Liste des sous-VIs
- Historique des révisions

Enregistrement sous forme de fichiers HTML ou RTF

Vous pouvez enregistrer la documentation du VI sous forme de fichiers HTML et RTF. Vous pouvez importer des fichiers HTML et RTF dans la plupart des applications de traitement de texte et utiliser des fichiers HTML

et RTF pour créer des fichiers d'aide compilés. Vous pouvez utiliser les fichiers HTML que LabVIEW génère pour afficher la documentation du VI sur le Web. Reportez-vous à la section *Création de vos propres fichiers d'aide* de ce chapitre afin d'obtenir plus de détails sur l'utilisation des fichiers HTML et RTF pour créer des fichiers d'aide. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation*, pour de plus amples informations sur la création de fichiers HTML et RTF par programmation.

Lorsque vous enregistrez de la documentation dans un fichier RTF, spécifiez si vous voulez créer un fichier approprié aux fichiers d'aide en ligne ou au traitement de texte. Pour le format de fichier d'aide, LabVIEW enregistre les graphiques dans des fichiers bitmap externes. Dans le cas du format de fichier de traitement de texte, LabVIEW intègre les graphiques dans le document. Pour les fichiers HTML, LabVIEW enregistre tous les graphiques séparément au format JPEG ou PNG.

Sélection des formats d'image pour les fichiers HTML

Lorsque vous enregistrez de la documentation dans un fichier HTML, vous pouvez sélectionner le format des fichiers image et la profondeur de couleurs.

Le format JPEG compresse correctement les graphiques mais peut perdre certains détails graphiques. Ce format convient particulièrement bien aux photos. La compression JPEG des lignes, des faces-avant et des diagrammes peut provoquer un certain flou dans les graphiques et les couleurs risquent d'être réparties inégalement. Les graphiques JPEG sont toujours des graphiques 24 bits. Si vous sélectionnez une profondeur de couleurs inférieure avec JPEG, noir et blanc par exemple, les graphiques s'enregistrent à la profondeur que vous avez demandée, mais le résultat est tout de même un graphique 24 bits.

Le format PNG compresse aussi les graphiques correctement, mais parfois moins bien que le format JPEG. Cependant, la compression PNG n'entraîne aucune perte de qualité au niveau des détails. De plus, elle accepte des graphiques 1 bit, 4 bits, 8 bits et 24 bits. Pour un nombre de bits par pixel inférieur, le graphique résultant compresse beaucoup mieux que le JPEG. Le format PNG remplace le format GIF (Graphics Interchange Format). Bien que le format PNG présente certains avantages par rapport aux formats JPEG et GIF, il n'est pas aussi bien supporté qu'eux par la plupart des navigateurs Web.

Le format GIF compresse également les graphiques correctement et est supporté par la plupart des navigateurs Web. Pour des questions de licence, LabVIEW n'enregistre pas les graphiques en fichiers GIF comprimés mais

pourrait le faire à l'avenir. Servez-vous d'un convertisseur de format d'image pour convertir les fichiers GIF non comprimés que LabVIEW enregistre en fichiers GIF comprimés. Vous pouvez également utiliser un convertisseur de format d'image pour convertir des graphiques PNG en graphiques GIF. Partez de graphiques PNG parce que ce sont des reproductions des graphiques originaux qui n'entraînent pas de perte. Modifiez le fichier HTML dans lequel vous avez enregistré la documentation du VI pour que les graphiques GIF portent l'extension .gif.

Conventions d'appellation des fichiers image

Lorsque vous enregistrez des fichiers HTML ou RTF qui comportent des images externes, LabVIEW enregistre les terminaux de commandes et d'indicateurs dans des fichiers image dont les noms sont compatibles. Si un VI a plusieurs terminaux du même type, LabVIEW crée un fichier image qui contient chaque terminal. Par exemple, si un VI a trois entrées d'entiers 32 bits signés, LabVIEW crée un seul fichier `ci32.jpg`.

Création de vos propres fichiers d'aide

Vous pouvez utiliser les fichiers HTML ou RTF que LabVIEW génère pour créer vos propres fichiers d'aide compilés. **(Windows)** Vous pouvez compiler les fichiers HTML individuels que LabVIEW génère en fichiers d'aide HTML.

Vous pouvez compiler les fichiers RTF que LabVIEW génère en un fichier **(Windows)** WinHelp, **(Macintosh)** QuickHelp ou **(UNIX)** HyperHelp.

Créez des liens entre des VIs et des fichiers HTML ou des fichiers d'aide compilés en sélectionnant **Fichier»Propriétés du VI...**, puis **Documentation** dans le menu déroulant **Catégorie**.

Impression des VIs

Vous pouvez utiliser les méthodes principales suivantes pour imprimer des VIs de LabVIEW :

- Sélectionnez **Fichier»Imprimer la fenêtre...** pour imprimer le contenu de la fenêtre active.
- Sélectionnez **Fichier»Imprimer...** pour effectuer une impression plus complète d'un VI, comprenant les informations sur la face-avant, le diagramme, les sous-VIs, les commandes, l'historique du VI et ainsi de suite. Reportez-vous à la section [Impression de la documentation](#) de ce

chapitre pour obtenir plus de détails sur l'utilisation de cette méthode d'impression des VIs.

- Utilisez les VIs de génération de rapport pour imprimer des rapports papier ou enregistrer des rapports HTML contenant les informations générées par le VI.
- Utilisez le VI Serveur pour imprimer par programmation n'importe quelle fenêtre ou documentation de VI à n'importe quel moment. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation*, pour obtenir plus de détails sur l'utilisation de cette méthode d'impression des VIs.

Imprimer la fenêtre active

Sélectionnez **Fichier»Imprimer la fenêtre...** pour imprimer le contenu de la fenêtre active de la face-avant ou du diagramme avec un nombre minimum de messages. LabVIEW imprime l'espace de travail de la fenêtre active mais n'est pas limité par la taille de la fenêtre active. LabVIEW n'imprime pas la barre de titre, la barre de menus, la barre d'outils ni les barres de défilement.

Sélectionnez **Fichier»Propriétés du VI...**, puis **Options d'impression** dans le menu déroulant **Catégorie**, pour définir davantage la façon dont LabVIEW imprime un VI lorsque vous sélectionnez **Fichier»Imprimer la fenêtre...** ou lorsque vous effectuez une impression par programmation. Reportez-vous à la section *Impression par programmation* de ce chapitre pour obtenir plus de détails sur l'impression de rapports par programmation.

Impression de rapports

Utilisez les VIs de génération de rapport, qui se trouvent dans la palette **Fonctions»Génération de rapport**, pour imprimer des rapports sur papier ou enregistrer des rapports HTML contenant des informations générées par le VI. Créez un rapport de base en utilisant le VI Rapport texte facile ou générez des rapports plus complexes à l'aide des autres VIs de génération de rapport.



Remarque La génération de rapports HTML est uniquement disponible dans les versions Systèmes de développement complet et professionnel de LabVIEW.

Vous pouvez utiliser les VIs de génération de rapport pour effectuer les tâches suivantes :

- Définir les en-têtes et les pieds de page du rapport.
- Définir la police, la taille, le style et la couleur du texte.
- Définir les marges et les tabulations.
- Déterminer l'emplacement d'affichage des informations par ligne et par page.
- Définir l'orientation du rapport (portrait ou paysage).
- Inclure dans un rapport du texte d'autres fichiers.
- Effacer les informations dans un rapport existant.
- Imprimer automatiquement un rapport papier ou enregistrer un rapport HTML.
- Ajouter du texte, des images ou des tables à un rapport.
- Fermer un rapport après son impression.

Impression par programmation

Utilisez les méthodes suivantes pour imprimer des VIs par programmation plutôt que de manière interactive au moyen des boîtes de dialogue **Fichier»Imprimer la fenêtre** et **Fichier»Imprimer** :

- Définissez un VI pour qu'il imprime sa face-avant à la fin de chaque exécution.
- Créez un sous-VI pour imprimer le VI.
- Utilisez le VI Serveur pour imprimer par programmation n'importe quelle fenêtre de VI ou documentation de VI à n'importe quel moment. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation*, pour obtenir plus de détails sur l'utilisation de cette méthode pour imprimer les VIs.

Impression à la fin de l'exécution

Sélectionnez **Exécution»Imprimer à la fin de l'exécution** pour imprimer un VI lorsque son exécution se termine. Vous pouvez aussi sélectionner **Fichier»Propriétés du VI...**, sélectionner **Options d'impression** dans le menu déroulant **Catégorie** et cocher la case **Impression automatique de la face-avant à la fin de chaque exécution**.

Lorsque vous sélectionnez cette option, LabVIEW imprime le contenu de la face-avant à chaque fois que le VI termine son exécution. Si vous utilisez le VI comme sous-VI, LabVIEW effectue l'impression lorsque ce sous-VI termine son exécution et avant que ce sous-VI ne retourne à l'appelant.

Utilisation d'un sous-VI pour effectuer une impression sélective à la fin de l'exécution

Dans certains cas, vous pouvez souhaiter qu'un VI n'effectue pas d'impression à la fin de chaque exécution : par exemple, si vous souhaitez que l'impression ne s'effectue que si l'utilisateur clique sur un bouton ou dans certaines circonstances, comme un échec de test, ou bien si vous souhaitez avoir plus de contrôle sur le format de sortie d'impression, ou n'imprimer qu'un sous-ensemble de commandes. Dans ce cas de figure, vous pouvez utiliser un sous-VI configuré pour être imprimé à la fin de l'exécution.

Créez un sous-VI et formatez la face-avant en fonction du format de sortie d'impression LabVIEW que vous souhaitez. Sélectionnez **Exécution» Imprimer à la fin de l'exécution** dans le sous-VI plutôt que dans le VI. Lorsque vous voulez imprimer, appelez le sous-VI et passez-lui les données à imprimer.

Techniques d'impression supplémentaires

Si les méthodes d'impression standard de LabVIEW ne répondent pas à vos besoins, vous pouvez aussi utiliser les techniques suivantes :

- Imprimez-les données ligne par ligne. Si vous avez une imprimante ligne par ligne connectée à un port série, utilisez les VIs de compatibilité série pour envoyer le texte à l'imprimante. Cette commande requiert une certaine maîtrise du langage machine de l'imprimante.
- Exportez les données dans d'autres applications, comme Microsoft Excel, enregistrez les données dans un fichier, et imprimez les à partir de l'autre application.
- **(Windows et UNIX)** Utilisez le VI System Exec.
- **(Macintosh)** Utilisez le VI AESend Print Document.
- **(Windows)** Utilisez ActiveX Automation pour faire imprimer des données par une autre application. Reportez-vous au chapitre 18, [ActiveX](#), pour plus d'informations sur ActiveX.

Personnalisation des VIs

Vous pouvez configurer les VIs et les sous-VIs pour les adapter aux besoins de vos applications. Par exemple, si vous avez l'intention d'utiliser un VI comme sous-VI qui requiert une entrée utilisateur, configurez le VI de telle sorte que sa face-avant s'affiche chaque fois que vous l'appellez.

Il existe de nombreuses manières de configurer un VI, soit de l'intérieur du VI lui-même, soit par programmation au moyen des fonctions de contrôle d'applications. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation*, pour obtenir plus de détails sur l'utilisation des fonctions de contrôle d'applications pour configurer plusieurs VIs de manière à ce qu'ils se comportent tous de la même manière.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir plus de détails sur la personnalisation des VIs.

Configuration de l'apparence et du comportement des VIs

Sélectionnez **Fichier»Propriétés du VI...** pour configurer l'apparence et le comportement d'un VI. Utilisez le menu déroulant **Catégorie**, situé en haut de la boîte de dialogue, pour choisir parmi plusieurs catégories d'options différentes, y compris les suivantes :

- **Général** : affiche le chemin de l'emplacement actuel où est enregistré un VI, son numéro de révision, son historique des révisions et tout changement apporté depuis le dernier enregistrement du VI. Vous pouvez également utiliser cette page pour éditer une icône.
- **Documentation** : utilisez cette page pour ajouter une description du VI et effectuer un lien vers un sujet d'un fichier d'aide. Reportez-vous à la section *Documentation des VIs* du chapitre 14, *Documenter et imprimer des VIs*, pour obtenir plus de détails sur les options de documentation.

- **Sécurité** : utilisez cette page pour verrouiller un VI ou le protéger par un mot de passe.
- **Apparence de la fenêtre** : utilisez cette page pour configurer divers paramètres de fenêtre.
- **Taille de la fenêtre** : utilisez cette page pour définir la taille de la fenêtre.
- **Exécution** : utilisez cette page pour configurer la façon dont s'exécute un VI. Ainsi, vous pouvez configurer un VI pour qu'il s'exécute à l'ouverture ou qu'il se mette en mode pause lorsqu'il est appelé en tant que sous-VI. Vous pouvez aussi configurer le VI pour qu'il s'exécute selon diverses priorités. Par exemple, s'il est capital qu'un VI s'exécute sans attendre la fin d'une autre opération, configurez son exécution sur la priorité critique de temps (la plus haute). Reportez-vous à la note d'application *Using LabVIEW to Create Multithreaded VIs for Maximum Performance and Reliability* pour obtenir plus de détails sur la création de VIs multithread.

Personnalisation des menus

Vous pouvez créer des menus personnalisés pour tous les VIs que vous construisez et vous pouvez choisir d'afficher ou de masquer les barres de menus des VIs. Affichez et masquez les barres de menus en sélectionnant **Fichier»Propriétés du VI...**, en sélectionnant **Apparence de la fenêtre** dans le menu déroulant **Catégorie**, en cliquant sur le bouton **Personnaliser...** et en cochant ou en décochant la case **Visualiser la barre de menus**.

Lorsque vous configurez un menu, vous le créez et vous lui fournissez le code du diagramme qui s'exécute quand l'utilisateur sélectionne les divers éléments du menu.



Remarque Les menus personnalisés apparaissent uniquement lors de l'exécution du VI.

Création de menus

Vous pouvez construire des menus personnalisés ou modifier les menus par défaut de LabVIEW de manière statique lorsque vous éditez le VI ou par programme lorsque vous l'exécutez. Lorsque vous sélectionnez **Édition»Menu d'exécution...** et que vous créez un menu dans la boîte de dialogue **Éditeur de menu**, LabVIEW crée un fichier de menu d'exécution (.xrtm) qui vous permet d'avoir une barre de menus personnalisée sur un VI plutôt que celle par défaut. Après la création et l'enregistrement du

fichier .rtm, vous pouvez conserver le même chemin relatif entre le VI et le fichier .rtm.

Utilisez la boîte de dialogue **Éditeur de menu** pour associer un fichier personnalisé .rtm à un VI. Lorsque le VI s'exécute, il charge le menu à partir du fichier .rtm. Vous pouvez aussi utiliser la boîte de dialogue **Éditeur de menu** pour construire des menus personnalisés soit avec des éléments de l'application, qui sont des éléments de menus fournis par LabVIEW dans le menu par défaut, soit avec des éléments utilisateur, qui sont des éléments de menus que vous ajoutez. LabVIEW définit le comportement des éléments de l'application mais vous contrôlez le comportement des éléments utilisateur avec le diagramme. Reportez-vous à la section *Gestion de la sélection des menus* de ce chapitre pour obtenir davantage d'informations sur la gestion des sélections d'éléments de menus utilisateur.

Utilisez la boîte de dialogue de l'Éditeur de menu pour personnaliser des menus lors de l'édition d'un VI. Utilisez les fonctions de menu situées dans la palette **Fonctions»Contrôle d'applications»Menu** pour personnaliser des menus par programmation lors de l'exécution. Ces fonctions vous permettent d'insérer, de supprimer et de modifier les attributs d'éléments utilisateur. Vous pouvez uniquement ajouter ou supprimer des éléments de l'application car LabVIEW définit le comportement et l'état de ceux-ci.

Gestion de la sélection des menus

Lorsque vous créez un menu personnalisé, vous assignez à chaque élément de menu un identificateur de chaîne unique, appelé tag, qui ne distingue pas les majuscules des minuscules. Lorsque l'utilisateur sélectionne un élément de menu, il récupère son tag par programmation au moyen de la fonction *Élément de menu sélectionné*. LabVIEW fournit un gestionnaire sur le diagramme pour chaque élément de menu en fonction de la valeur du tag de chaque élément de menu. Le gestionnaire est la combinaison d'une boucle While et d'une structure Condition qui vous permet de déterminer le menu éventuellement sélectionné et d'exécuter le code approprié.

Après avoir construit un menu personnalisé, construisez, dans le diagramme, une structure Condition qui va exécuter ou gérer chaque élément du menu personnalisé. Cette procédure s'appelle la gestion d'éléments de menu sélectionnés. LabVIEW gère tous les éléments de l'application implicitement.

Dans la figure15-1, la fonction Élément de menu sélectionné lit l'élément de menu sélectionné par l'utilisateur et transfère cet élément de menu dans la structure Condition où il va s'exécuter.

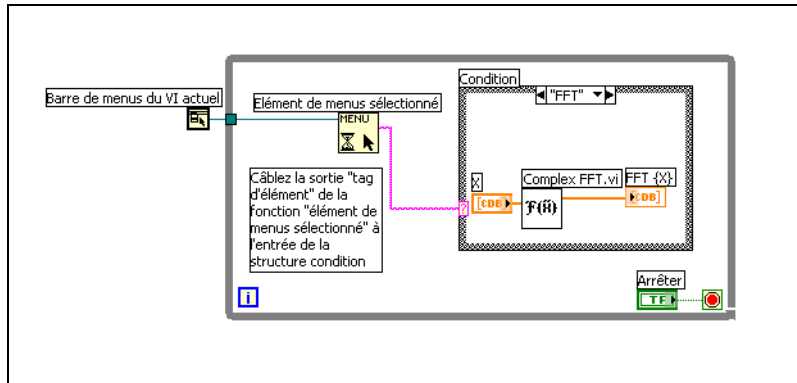


Figure 15-1. Diagramme utilisant la gestion des menus

Si vous savez qu'un élément de menu particulier met beaucoup de temps à s'exécuter, câblez une commande booléenne à l'entrée **menu** de la fonction Élément de menu sélectionné et définissez la commande booléenne sur VRAI pour désactiver la barre de menus, afin que l'utilisateur ne puisse rien sélectionner d'autre dans le menu pendant que LabVIEW exécute l'élément de menu. Câblez une valeur VRAI à la fonction Activer le repérage des menus pour activer la barre de menus une fois que LabVIEW a exécuté l'élément de menu.

Contrôle des VIs par programmation

Vous pouvez accéder au VI Serveur par l'intermédiaire des diagrammes, de la technologie ActiveX et du protocole TCP pour communiquer avec des VIs et d'autres instances de LabVIEW, afin de pouvoir contrôler par programmation les VIs et LabVIEW. Vous pouvez effectuer les opérations de VI Serveur sur un ordinateur local ou sur un réseau à distance.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour de plus amples informations sur le contrôle des VIs par programmation.

Caractéristiques du VI Serveur

Utilisez le VI Serveur pour effectuer les opérations de programmation suivantes :

- Appeler un VI à distance.
- Configurer une instance de LabVIEW pour qu'elle joue le rôle d'un serveur qui exporte des VIs que vous pouvez appeler à partir d'autres instances de LabVIEW sur le Web. Par exemple, si vous avez une application d'acquisition de données qui acquiert et enregistre des données à distance, vous pouvez échantillonner ces données à partir de votre ordinateur local. En changeant vos préférences LabVIEW, vous pouvez rendre certains VIs accessibles sur le Web, ce qui rend le transfert des données les plus récentes aussi facile que par un appel à un sous-VI. Le VI Serveur gère le réseau en détails. Le VI Serveur peut passer d'une plate-forme à une autre, ce qui permet au client d'utiliser une plate-forme différente de celle du serveur.
- Éditer les propriétés d'un VI et celles de LabVIEW. Ainsi, vous pouvez déterminer dynamiquement l'emplacement de la fenêtre d'un VI ou faire défiler une face-avant pour la rendre en partie visible. Vous pouvez aussi enregistrer toutes les modifications effectuées dans le disque par programmation.

- Mettre à jour les propriétés de plusieurs VIs plutôt que de le faire manuellement via la boîte de dialogue **Fichier»Propriétés du VI...** pour chaque VI.
- Extraire des informations d'une instance de LabVIEW, comme le numéro de version et l'édition. Vous pouvez aussi extraire des informations sur l'environnement, notamment sur la plate-forme sur laquelle LabVIEW est en cours d'exécution.
- Charger dynamiquement des VIs en mémoire quand un autre VI a besoin de les appeler, plutôt que de charger tous les sous-VIs quand vous ouvrez un VI.
- Créer une architecture modulaire pour l'application afin de pouvoir lui ajouter des fonctions après sa répartition entre les clients. Par exemple, considérons un ensemble de VIs de filtrage des données qui utilisent tous les mêmes paramètres. En concevant l'application de telle sorte qu'elle charge dynamiquement ces VIs à partir d'un répertoire modulaire, vous pouvez livrer l'application avec un ensemble partiel de ces VIs et rendre un plus grand nombre d'options de filtrage disponibles aux utilisateurs en plaçant les nouveaux VIs de filtrage dans le répertoire modulaire.

Construire des applications du VI Serveur

Le modèle de programmation pour les applications du VI Serveur est basé sur les refnums. Les refnums sont aussi utilisés dans les E/S sur fichiers, les connexions réseau et d'autres objets de LabVIEW. Reportez-vous à la section [Références à des objets ou à des applications](#) du chapitre 4, [Construction de la face-avant](#), pour obtenir plus de détails sur les refnums.

Habituellement, vous ouvrez un refnum vers une instance de LabVIEW ou vers un VI. Vous pouvez alors utiliser le refnum comme paramètre dans d'autres VIs. Les VIs reçoivent (lisent) ou définissent (écrivent) des propriétés, exécutent des méthodes, ou chargent dynamiquement un VI référencé. Pour finir, vous fermez le refnum, ce qui libère le VI référencé de la mémoire.

Utilisez les fonctions et les nœuds suivants, situés dans la palette **Fonctions»Contrôle d'applications** pour construire des applications de type VI Serveur :

- **Ouvrir une référence d'application** : ouvre un refnum pour l'application locale ou l'application à distance à laquelle vous avez accès via le serveur ou pour accéder à une instance de LabVIEW

distante. Utilisez la fonction Ouvrir une référence de VI pour accéder à un VI sur un ordinateur local ou un ordinateur à distance.

- **Nœud de propriété** : obtient et définit les propriétés du VI ou de l'application. Reportez-vous à la section *Nœuds de propriété* de ce chapitre pour obtenir plus de détails sur les propriétés.
- **Nœud de méthode** : appelle les méthodes d'un VI ou d'une application. Reportez-vous à la section *Nœuds de méthode* de ce chapitre pour obtenir plus de détails sur les méthodes.
- **Nœud d'appel par référence** : appelle dynamiquement le VI référencé.
- **Fermer la référence à un objet LV** : ferme le refnum du VI ou l'application à laquelle vous avez accédé au moyen du VI Serveur.

Références d'application et du VI

Vous accédez aux fonctions du VI Serveur via des références à deux classes principales d'objets : l'objet application et l'objet VI. Après avoir créé une référence à l'un de ces objets, vous pouvez transférer la référence à un VI ou à une fonction qui effectue une opération sur l'objet.

Un refnum d'application fait référence à une instance de LabVIEW locale ou distante. Vous pouvez utiliser les propriétés et les méthodes d'application pour modifier les préférences LabVIEW et renvoyer les informations du système. Un refnum de VI fait référence à un VI dans l'instance de LabVIEW.

Avec un refnum se référant à une instance de LabVIEW, vous pouvez extraire des informations sur l'environnement LabVIEW, par exemple sur la plate-forme sur laquelle LabVIEW est en cours d'exécution, le numéro de version ou la liste de tous les VIs actuellement en mémoire. Vous pouvez aussi définir des informations, comme le nom de l'utilisateur courant ou la liste des VIs exportés dans d'autres instances de LabVIEW.

Quand vous obtenez un refnum pour un VI, vous chargez le VI dans la mémoire. Une fois que vous avez obtenu le refnum, le VI reste en mémoire jusqu'à ce que vous fermiez le refnum. Si plusieurs refnums coexistent pour un VI ouvert, le VI reste en mémoire jusqu'à ce que vous fermiez tous les refnums du VI. Avec un refnum de VI, vous pouvez mettre à jour toutes les propriétés du VI disponibles dans la boîte de dialogue **Fichier»Propriétés du VI...**, de même que les propriétés dynamiques telles que la position de la fenêtre de la face-avant. Vous pouvez aussi imprimer le VI par programmation, l'enregistrer à un autre endroit, et exporter et importer ses chaînes pour les traduire dans une autre langue.

Édition des paramètres de l'application et du VI

Utilisez le VI Serveur pour obtenir et définir des paramètres d'application et de VI au moyen des nœuds de propriété et des nœuds de méthode.

De nombreux paramètres d'application et de VI ne peuvent être obtenus et définis qu'au moyen des nœuds de propriété et des nœuds de méthode.

Reportez-vous à `exemples\viserver` si vous voulez consulter des exemples d'utilisation des propriétés et des méthodes de classe d'application et de VI.

Nœuds de propriété

Utilisez le nœud de propriété pour obtenir et définir diverses propriétés d'application ou d'un VI. Sélectionnez les propriétés à partir du nœud en vous servant de l'outil Doigt pour cliquer sur le terminal de propriété ou en cliquant sur la zone blanche du nœud avec le bouton droit de la souris et en sélectionnant **Propriétés** dans le menu local.

Vous pouvez lire ou écrire plusieurs propriétés au moyen d'un nœud unique. Utilisez l'outil Flèche pour redimensionner le nœud de propriété afin d'y ajouter de nouveaux terminaux. Une petite flèche de direction à droite de la propriété indique une propriété à lire. Une petite flèche de direction à gauche de la propriété indique une propriété à écrire. Cliquez sur la propriété avec le bouton droit de la souris et sélectionnez **Changer en lecture** ou **Changer en écriture** dans le menu local pour changer l'opération de la propriété.

Le nœud s'exécute de haut en bas. Le nœud de propriété ne s'exécute pas si une erreur survient avant l'exécution ; par conséquent, vérifiez toujours s'il existe une possibilité d'erreur. Si une erreur survient dans une propriété, LabVIEW ignore les propriétés restantes et renvoie une erreur. Le cluster d'**erreur de sortie** contient des informations sur la propriété qui a causé l'erreur.

Nœuds de propriété liés implicitement

Lorsque vous créez un nœud de propriété à partir d'un objet de la face-avant en cliquant sur l'objet avec le bouton droit de la souris et en sélectionnant **Créer»Nœud de propriété** dans le menu local, LabVIEW crée un nœud de propriété sur le diagramme qui est implicitement lié à l'objet. Dans la mesure où ces nœuds de propriété sont implicitement liés à l'objet à partir duquel ils ont été créés, ils n'ont pas d'entrée de **refnum**, et vous n'avez donc pas à câbler le nœud de propriété à la commande ou au refnum de

commande. Reportez-vous à la section *Contrôle des objets de la face-avant* de ce chapitre pour plus d'informations sur les refnoms de commandes.

Nœuds de méthode

Utilisez le nœud de méthode pour effectuer des actions ou méthodes sur une application ou un VI. À la différence du nœud de propriété, un seul nœud de méthode n'exécute qu'une seule méthode sur une application ou un VI. Sélectionnez une méthode à partir du nœud en vous servant de l'outil Doigt pour cliquer sur le terminal de méthode ou en cliquant sur la zone blanche du nœud avec le bouton droit de la souris et en sélectionnant **Méthodes** dans le menu local.

Le nom de la méthode est toujours le premier terminal de la liste de paramètres dans le nœud de méthode. Si la méthode retourne une valeur, le terminal de la méthode affiche la valeur retournée. Sinon, le terminal de la méthode n'affiche aucune valeur.

Le nœud de méthode liste les paramètres de haut en bas, avec le nom de la méthode en haut et les paramètres optionnels grisés dans le bas.

Manipulation des propriétés et des méthodes de classe d'application

Vous pouvez obtenir ou définir des propriétés dans une instance locale de LabVIEW, ou dans une instance à distance, effectuer des méthodes dans LabVIEW, ou bien les deux. La figure 16-1 montre comment afficher tous les VIs en mémoire sur un ordinateur local dans un tableau de chaînes sur la face-avant.

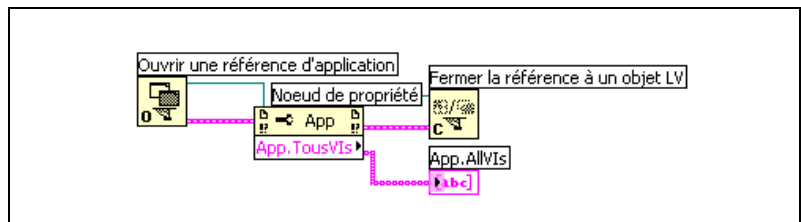


Figure 16-1. Affichage de tous les VIs en mémoire sur un ordinateur local

Pour rechercher les VIs en mémoire sur un ordinateur à distance, câblez une commande de chaîne à l'entrée du **nom de la machine** de la fonction Ouvrir une référence d'application, comme illustré à la figure 16-2, et entrez l'adresse IP ou le nom de domaine. Vous devez aussi sélectionner la propriété **VIs exportés en mémoire** parce que la propriété **VIs en**

mémoire utilisée à la figure 16-1 ne s'applique qu'aux instances locales de LabVIEW.

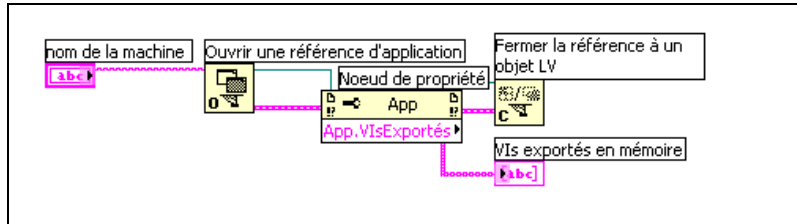


Figure 16-2. Affichage de tous les VIs en mémoire sur un ordinateur à distance

Manipulation des propriétés et des méthodes de classe de VI

Vous pouvez obtenir ou définir les propriétés d'un VI, effectuer des méthodes sur un VI, ou bien les deux. Dans la figure 16-3, LabVIEW réinitialise les objets de la face-avant d'un VI à leur valeur par défaut au moyen du nœud de méthode. La face-avant ouvre et affiche les valeurs par défaut au moyen du nœud de propriété.

Avant de pouvoir accéder aux propriétés et aux méthodes d'un VI, vous devez créer un refnum vers ce VI au moyen de la fonction Ouvrir une référence de VI. Utilisez le nœud de méthode pour appeler une méthode sur un VI.

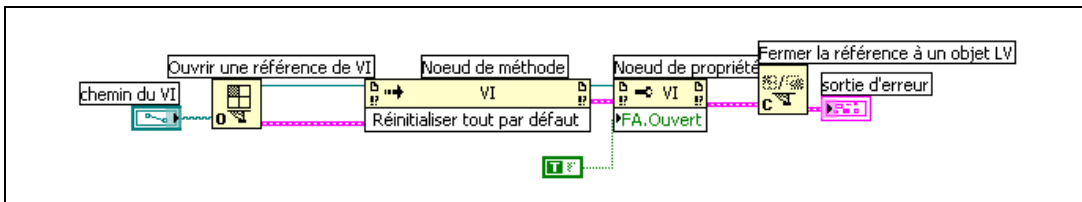


Figure 16-3. Utilisation des nœuds de méthode et de propriété de classe de VI

Une fois que vous avez câblé la fonction Ouvrir une référence de VI au nœud de méthode, vous pouvez accéder à toutes les méthodes de classe de VI.

Le nœud de propriété fonctionne de la même manière que le nœud de méthode. Une fois que vous avez câblé un refnum de VI au nœud de propriété, vous pouvez accéder à toutes les propriétés de classe de VI.

Manipulation des propriétés et des méthodes de classe d'application et de VI

Pour certains VIs, vous devez accéder aux propriétés et aux méthodes de classe d'application et de VI en même temps. Vous devez ouvrir et fermer les refnoms de classe d'application et de VI séparément, comme l'illustre la figure 16-4.

La figure 16-4 montre comment déterminer quels VIs sont en mémoire dans un ordinateur local et comment afficher le chemin de chacun des VIs sur la face-avant. Pour rechercher tous les VIs en mémoire, vous devez accéder à une propriété de classe d'application. Pour déterminer les chemins de ces VIs, vous devez accéder à une propriété de classe de VI. Le nombre de VIs en mémoire détermine le nombre de fois que la boucle For s'exécute. Placez les fonctions Ouvrir une référence de VI et Fermer la référence à l'objet LV à l'intérieur de la boucle For, parce que vous avez besoin d'un refnum de VI pour chaque VI en mémoire. Ne fermez pas le refnum d'application tant que la boucle For n'a pas terminé d'extraire tous les chemins des VIs.

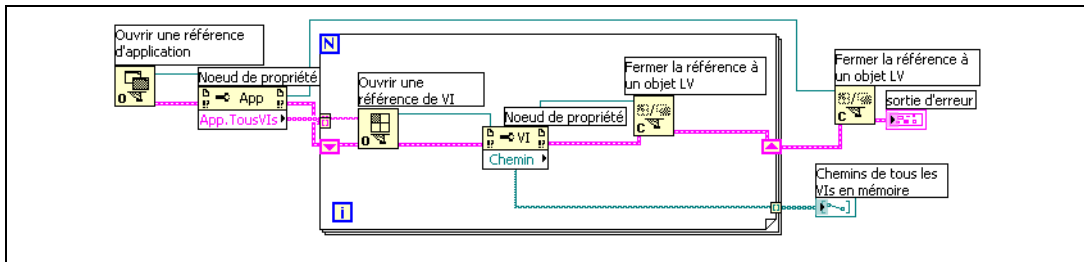


Figure 16-4. Utilisation des propriétés et méthodes d'application et de classe de VI

Charger et appeler les VIs de façon dynamique

Vous pouvez charger des VIs de façon dynamique au lieu d'utiliser les appels de sous-VIs liés de façon statique. Un sous-VI lié de façon statique est un sous-VI qui se place directement dans le diagramme d'un VI appelant. Il se charge en même temps que le VI appelant.

À la différence des sous-VIs liés de façon statique, les sous-VIs chargés de façon dynamique ne se chargent pas tant que le VI appelant n'a pas appelé le sous-VI. Si votre VI appelant est grand, vous pouvez gagner du temps et de la mémoire en chargeant le sous-VI de façon dynamique, vu que le sous-VI ne se charge pas tant que le VI appelant n'en a pas besoin et que vous pouvez le libérer de la mémoire à la fin de l'opération.

Nœuds d'appel par référence et refnums de VI de type strict

Utilisez le nœud d'appel par référence pour appeler les VIs de façon dynamique.

Le nœud d'appel par référence a besoin d'un refnum de VI de type strict. Le refnum de VI de type strict identifie le connecteur du VI que vous appelez. Il ne crée pas d'association permanente au VI et ne contient pas d'autre information sur le VI, comme le nom ou la position. Vous pouvez câbler les entrées et les sorties du nœud d'appel par référence de la même manière que vous câblez les autres VIs.

La figure 16-5 montre comment se servir du nœud d'appel par référence pour appeler le VI de réponse en fréquence de façon dynamique. Avec le nœud d'appel par référence, vous devez utiliser aussi les fonctions Ouvrir une référence de VI et Fermer la référence à l'objet LV, similaires aux fonctions que vous utilisez pour le nœud de propriété et le nœud de méthode. Cependant, avec le nœud d'appel par référence, vous devez câbler un refnum de VI de type strict à l'entrée du **Refnum du VI Spécificateur du type** de la fonction Ouvrir une référence de VI. Câblez un refnum de VI de type strict en sélectionnant **Commandes»Refnum»Refnum de VI** et en plaçant le refnum sur la face-avant. Cliquez sur le refnum avec le bouton droit de la souris et sélectionnez **Sélectionner une classe de VI Serveur»Parcourir...** dans le menu local. La boîte de dialogue **Ouvrir un VI** vous invite à entrer un nom de VI.

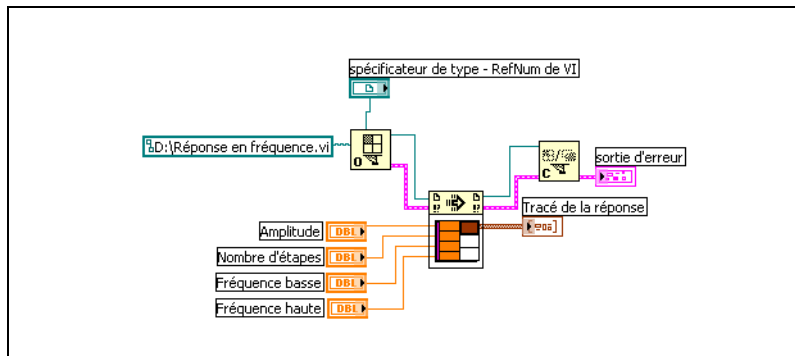


Figure 16-5. Utilisation du nœud d'appel par référence

Le VI que vous spécifiez pour les refnums de type strict ne fournit que les informations du connecteur. Cela signifie qu'aucune association permanente n'est créée entre le refnum et le VI. Évitez, en particulier, de confondre l'action de sélectionner le connecteur du VI avec celle d'obtenir

un refnum vers le VI sélectionné. Vous spécifiez un VI particulier à l'aide de l'entrée **chemin du VI** de la fonction Ouvrir une référence de VI.

Édition et exécution de VIs sur des ordinateurs à distance

Un aspect important des refnums de VIs et des refnums d'application est leur transparence réseau. Cette transparence vous permet d'ouvrir des refnums vers des objets sur des ordinateurs à distance de la même manière que vous ouvrez des refnums vers ces objets sur votre ordinateur.

Après avoir ouvert un refnum vers un objet à distance, vous pouvez le traiter exactement de la même manière qu'un objet local, à quelques exceptions près. Pour des opérations sur un objet à distance, le VI Serveur envoie des informations sur l'opération à tout le réseau et retourne les résultats. L'application est presque identique, qu'il s'agisse d'une opération à distance ou d'une opération locale.

Contrôle des objets de la face-avant

Utilisez les refnums de commande, situés sur les palettes **Commandes» Refnum** et **Commandes»Commandes classiques»Refnum**, pour transférer les références aux objets de la face-avant à d'autres VIs. Une fois que vous avez transféré un refnum de commande à un sous-VI, utilisez les nœuds de propriété et les nœuds de méthode pour lire et configurer les propriétés et pour appeler les méthodes d'un objet référencé de la face-avant. Reportez-vous à la section *Structures Condition, Séquence et Événement* du chapitre 8, *Boucles For et While et Structures Condition, Séquence et Événement*, pour obtenir des informations sur l'utilisation d'événements afin de contrôler par programmation le comportement du diagramme au moyen d'objets de la face-avant.

Refnums de commande de type strict et refnums de commande partiellement spécifiés

Les refnums de commande de type strict n'acceptent que les refnums de commande de même type de données. Par exemple, si le type d'un refnum de commande de type strict est une glissière d'entiers 32 bits, vous pouvez câbler une glissière d'entiers 32 bits, une glissière d'entiers 8 bits ou une glissière de scalaires double précision au terminal du refnum de commande, mais pas une glissière d'un cluster d'entiers 32 bits.

Les refnums de commande que vous créez à partir d'une commande sont de type strict par défaut. Une étoile rouge dans le coin inférieur gauche du refnum de commande sur la face-avant indique que le refnum de commande est de type strict. Dans le diagramme, (`strict`) s'affiche dans le nœud de propriété ou le nœud de méthode câblé au terminal du refnum de commande pour indiquer que le refnum de commande est de type strict.



Remarque Les actions mécaniques d'armement étant incompatibles avec les refnums de commande de type strict, les commandes booléennes à action mécanique d'armement produisent des refnums de commande partiellement spécifiés.

Les refnums de commande partiellement spécifiés sont plus flexibles au niveau du type de données qu'ils acceptent. Ainsi, si le type d'un refnum de commande partiellement spécifié est une glissière, vous pouvez câbler une glissière d'entiers 32 bits, une glissière de simple précision ou un cluster de glissières d'entiers 32 bits au terminal du refnum de commande. Si le type d'un refnum de commande partiellement spécifié est la commande, vous pouvez câbler un refnum de commande de tout type de commande au terminal du refnum de commande.



Remarque Lorsque vous câblez un nœud de propriété à un terminal de refnum de commande partiellement spécifié, la propriété Valeur produit des données de type variant qui doivent parfois être converties avant de pouvoir être utilisées. La propriété Données d'historique n'est disponible que si le refnum de graphe déroulant est de type strict. Reportez-vous à la section [Manipulation de données variant du chapitre 5](#), [Construction du diagramme](#), pour obtenir plus de détails sur les données variant.

Mise en réseau dans LabVIEW

Les VIs peuvent communiquer ou être en réseau, avec d'autres procédés, y compris ceux qui s'exécutent sur d'autres applications ou sur des ordinateurs à distance. Utilisez les caractéristiques de mise en réseau de LabVIEW pour effectuer les tâches suivantes :

- Partager des données en direct avec d'autres VIs qui s'exécutent sur un réseau utilisant la technologie DataSocket de National Instruments.
- Publier des images de face-avant et de la documentation de VI sur le Web.
- Construire des VIs qui communiquent avec d'autres applications et d'autres VIs au moyen de protocoles de bas niveau, tels que TCP, UDP, DDE, Apple Events, et PPC Toolbox.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir plus de détails sur la mise en réseau.

Choix entre E/S sur fichiers, VI Serveur, ActiveX et mise en réseau

La mise en réseau n'est peut-être pas la meilleure solution pour votre application. Si vous voulez créer un fichier contenant des données lisibles par d'autres VIs et d'autres applications, utilisez les VIs et les fonctions d'E/S sur fichiers. Reportez-vous au chapitre 13, *E/S sur fichiers*, pour obtenir de plus amples informations à propos de l'utilisation des VIs et des fonctions d'E/S sur fichiers.

Si vous voulez contrôler d'autres VIs, utilisez le VI Serveur. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation*, si vous souhaitez plus d'informations sur la manipulation des VIs et des autres applications LabVIEW sur des ordinateurs locaux ou à distance.

(Windows) Pour accéder aux fonctions des nombreuses applications de Microsoft, telles que l'intégration d'un graphe waveform dans fichier

tableur d'Excel, utilisez les VIs et les fonctions ActiveX. Reportez-vous au chapitre 18, *ActiveX*, si vous voulez en savoir plus sur la manière d'accéder aux applications ActiveX et de permettre aux applications ActiveX d'accéder à LabVIEW.

LabVIEW comme client et serveur réseau

Vous pouvez utiliser LabVIEW en tant que client, pour souscrire aux données et utiliser les fonctions d'autres applications, ou en tant que serveur, pour rendre les fonctions de LabVIEW accessibles par d'autres applications. Reportez-vous au chapitre 16, *Contrôle des VIs par programmation*, si vous souhaitez plus d'informations sur la manipulation des VIs et des autres applications LabVIEW sur des ordinateurs locaux et à distance. Vous contrôlez les VIs en accédant aux propriétés, au moyen d'un nœud de propriété, et en appelant les méthodes, au moyen d'un nœud de méthode.

Avant de pouvoir accéder aux propriétés et d'appeler les méthodes d'une autre application, vous devez établir le protocole de réseau que vous utilisez pour accéder aux propriétés et aux méthodes. Les protocoles que vous pouvez utiliser comprennent HTTP et TCP/IP. Vous sélectionnez un protocole en fonction de l'application. Par exemple, le protocole HTTP convient parfaitement à la publication sur le Web, mais vous ne pouvez pas l'utiliser pour construire un VI qui attend des données créées par un autre VI. Pour effectuer cette opération, utilisez le protocole TCP.

Reportez-vous à la section *Applications de communication de bas niveau* dans ce chapitre si vous souhaitez plus d'informations sur les protocoles de communication acceptés par LabVIEW.

(Windows) Reportez-vous au chapitre 18, *ActiveX*, si vous souhaitez plus d'informations sur l'utilisation de la technologie ActiveX avec LabVIEW comme serveur ou client ActiveX.

Utilisation de la technologie DataSocket

Utilisez la technologie DataSocket de National Instruments pour partager des données en direct avec d'autres VIs et d'autres applications, comme ComponentWorks de National Instruments, sur le Web ou sur votre ordinateur local. DataSocket assemble des protocoles de communication pour la mesure et l'automatisation de la même façon qu'un navigateur Web assemble diverses technologies Internet.

La technologie DataSocket donne accès à plusieurs mécanismes d'entrée et de sortie à partir de la face-avant par l'intermédiaire de la boîte de dialogue **Connexion DataSocket**. Cliquez avec le bouton droit sur un objet de la face-avant et sélectionnez **Opérations sur les données»Connexion DataSocket...** dans le menu local pour afficher la boîte de dialogue **Connexion DataSocket**. Vous publiez (écrivez) ou vous souscrivez à (lisez) des données en spécifiant une URL, de la même manière que vous en spécifiez une dans un navigateur Web.

Par exemple, pour partager les données de l'indicateur d'un thermomètre sur la face-avant avec d'autres ordinateurs sur le Web, publiez les données du thermomètre en spécifiant une URL dans la boîte de dialogue **Connexion DataSocket**. Les utilisateurs des autres ordinateurs souscrivent aux données en plaçant un thermomètre sur leur face-avant et en sélectionnant l'URL dans la boîte de dialogue **Connexion DataSocket**. Reportez-vous à la section *Utilisation de DataSocket sur la face-avant* plus loin dans ce chapitre pour obtenir de plus amples informations sur l'utilisation de la technologie DataSocket sur la face-avant.

Reportez-vous à la brochure *Intégration d'Internet dans votre système de mesure* pour de plus amples informations sur la technologie DataSocket. Cette brochure est disponible sous forme de fichier PDF sur le CD d'installation, dans le répertoire `manuals` ou à partir du site Web de National Instruments à l'adresse `ni.com`.

Spécification d'une URL

Les URL utilisent des protocoles de communication, tels que `dstp`, `ftp` et `fichier`, pour transférer des données. Le protocole que vous spécifiez dans une URL dépend du type de données que vous voulez publier et de la manière dont votre réseau est configuré.

Vous pouvez utiliser les protocoles suivants lorsque vous publiez ou souscrivez à des données à l'aide de DataSocket :

- **Protocole DataSocket Transport (dstp)** : le protocole natif pour les connexions DataSocket. Lorsque vous utilisez ce protocole, le VI communique avec le Serveur DataSocket. Vous devez fournir un tag nommé aux données qui s'ajoutent à l'URL. La connexion DataSocket utilise le tag nommé pour adresser un élément de données particulier au Serveur Datasocket. Pour utiliser ce protocole, vous devez exécuter un Serveur DataSocket.
- **(Windows) OLE for Process Control (opc)** : conçu spécialement pour partager des données de production en temps réel, telles que les

données générées par des opérations d'automatisation industrielles. Pour utiliser ce protocole, vous devez exécuter un serveur OPC.

- **(Windows) logos** : une technologie interne à National Instruments pour transmettre des données entre le réseau et votre ordinateur local.
- **Protocole de transfert de fichiers (ftp)** : vous pouvez utiliser ce protocole pour spécifier un fichier à partir duquel vous voulez lire des données.
- **fichier** : vous pouvez utiliser ce protocole pour fournir un lien vers un fichier local ou un réseau qui contient des données.

La table 17-1 fournit des exemples d'URL pour chaque protocole.

Tableau 17-1. Exemples d'URL DataSocket

URL	Exemple
dstp	dstp://servername.com/numericdata, où numericdata est le tag nommé
opc	opc:\National Instruments.OPCTest\item1 opc:\\machine\National Instruments.OPCModbus\Modbus Demo Box.4:0 opc:\\machine\National Instruments.OPCModbus\Modbus Demo Box.4:0?updaterate=100&deadband=0.7
logos	logos://computer_name/process/data_item_name
ftp	ftp://ftp.ni.com/datasocket/ping.wav
fichier	file:ping.wav file:c:\mydata\ping.wav file:\\machine\mydata\ping.wav

Utilisez les URL dstp, opc et logos pour partager des données en direct, étant donné que ces protocoles peuvent mettre à jour les commandes ni les indicateurs à distance ou locaux. Utilisez les URL ftp et fichier pour lire des données dans des fichiers, parce que ces protocoles ne peuvent pas mettre à jour les commandes et les indicateurs à distance ou locaux.

Reportez-vous au fichier `examples/comm/datasocketx.llb` pour obtenir des exemples d'utilisation des connexions DataSocket.

Formats de données supportés par DataSocket

Utilisez DataSocket pour publier les données suivantes et pour y souscrire :

- **Texte brut** : utilisez du texte brut pour livrer une chaîne à un indicateur de chaîne.
- **Texte à onglets** : utilisez du texte à onglets, comme celui d'un tableau, pour publier des données sous forme de tableaux. LabVIEW interprète le texte à onglets en tant que tableau de données.
- **Données wav** : utilisez les données .wav pour publier un son dans un VI ou dans une fonction.
- **Données variant** : utilisez les données variant pour souscrire aux données d'une autre application, telles que celles d'une commande ComponentWorks.

Utilisation de DataSocket sur la face-avant

Utilisez les connexions DataSocket de la face-avant pour publier ou souscrire en direct à des données d'un objet de la face-avant. Vous publiez des données lorsque vous partagez les données d'une face-avant avec d'autres utilisateurs. Lorsque les utilisateurs extraient les données publiées et les visualisent sur leur face-avant, ils souscrivent aux données.

Les connexions DataSocket diffèrent des connexions aux serveurs Web et des connexions ActiveX dans la mesure où vous pouvez utiliser ces connexions DataSocket directement à partir de la face-avant, sans avoir à programmer de diagramme. Chaque commande ou indicateur de la face-avant peut publier des données, ou souscrire à des données, via sa propre connexion DataSocket. Les connexions DataSocket de la face-avant publient uniquement les données et non pas le graphique d'une commande ; les VIs qui souscrivent aux données via une connexion DataSocket peuvent alors effectuer leurs propres opérations sur ces données.

Vous pouvez définir la valeur d'une commande de la face-avant directement sur la face-avant et publier ensuite les données, ou encore construire un diagramme, câbler la sortie d'un VI ou d'une fonction à un indicateur et publier les données à partir de cet indicateur. La liste suivante décrit les scénarii habituels d'utilisation des connexions DataSocket avec les commandes et les indicateurs :

- Publiez une valeur à partir d'une commande de la face-avant pour manipuler une commande et publier les données afin que les autres utilisateurs puissent y souscrire via une commande ou un indicateur. Par exemple, si vous placez sur la face-avant un bouton rotatif qui augmente ou réduit la température, l'utilisateur d'un autre ordinateur

peut souscrire aux données et les utiliser via une commande câblée à un sous-VI ou à une fonction, ou bien il peut visualiser les données dans un indicateur. Un utilisateur ne peut pas manipuler une commande de face-avant sur son VI pendant que la commande souscrit à des données à partir de la connexion DataSocket.

- Publiez une valeur qui apparaît dans un indicateur de la face-avant, afin qu'un autre utilisateur puisse souscrire aux données et visualiser les données dans une commande ou un indicateur sur sa face-avant ou utiliser les résultats en tant que données dans une commande câblée à une entrée dans un sous-VI ou une fonction. Par exemple, un VI qui calcule la température moyenne et l'affiche dans un thermomètre sur la face-avant peut publier les données sur la température.
- Souscrivez à une valeur à partir d'un indicateur de la face-avant pour visualiser les données d'un indicateur de la face-avant de votre VI qui s'affiche sur une commande ou sur un indicateur de la face-avant d'un autre VI.
- Souscrivez à une valeur à partir d'une commande de la face-avant pour visualiser les données d'une commande de la face-avant de votre VI qui s'affiche sur une commande ou sur indicateur de la face-avant d'un autre VI. Si vous souscrivez aux données via une commande, vous pouvez utiliser les données dans votre VI en câblant la commande à l'entrée d'un sous-VI ou d'une fonction.
- Publiez à partir d'une commande de face-avant et souscrivez à une commande de face-avant pour permettre aux utilisateurs de manipuler une commande sur la face-avant de votre VI à partir des faces-avant de leurs VIs. Quand un utilisateur modifie la valeur de la commande dans sa face-avant, la connexion DataSocket publie la nouvelle valeur dans la commande de la face-avant de votre VI. Si vous manipulez ensuite la valeur de la commande sur votre face-avant, votre VI publie la valeur sur la face-avant des autres utilisateurs.

Les objets de la face-avant qui souscrivent aux données ne doivent pas être les mêmes que ceux qui publient les données. Cependant, les objets de la face-avant doivent être du même type de données ou, dans le cas de données numériques, ils doivent contraindre ces dernières. Par exemple, vous pouvez utiliser un indicateur numérique de votre VI pour visualiser les données que le thermomètre d'un autre VI génère. Le thermomètre peut être un flottant et l'indicateur numérique peut être un entier.

Les connexions DataSocket de la face-avant servent essentiellement à partager des données en direct. Pour lire des données dans des fichiers locaux, les serveurs ftp ou les serveurs Web utilisent la fonction DataSocket Lire située dans la palette **Fonctions»Communication»DataSocket**,

les VIs d'E/S sur fichiers et les fonctions situées dans la palette **Fonctions»E/S sur fichiers** ou les fonctions de contrôle d'application situées dans la palette **Fonctions»Contrôle d'applications**. National Instruments vous recommande de ne publier que du texte brut et des données de type variant pour vos mises à jour en direct de face-avant.

Lecture et écriture de données en direct par l'intermédiaire du diagramme

À partir du diagramme, vous pouvez lire ou écrire des données par programmation en vous servant des fonctions DataSocket situées dans la palette **Fonctions»Communication»DataSocket**.

Utilisez la fonction DataSocket Écrire pour écrire des données en direct au moyen d'un programme par l'intermédiaire d'une connexion DataSocket. La figure 17-1 explique comment écrire une valeur numérique.

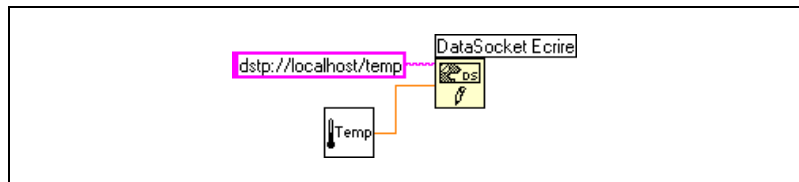


Figure 17-1. Publication de données à l'aide de DataSocket Écrire

La fonction DataSocket Écrire est polymorphe, ce qui vous permet de câbler la plupart des types de données à l'entrée **données**. Reportez-vous à la section *Vis et fonctions polymorphes* au chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur les VIs et les fonctions polymorphes.

Utilisez la fonction DataSocket Lire pour lire des données en direct au moyen d'un programme à partir d'une connexion DataSocket. La figure 17-2 montre comment lire des données et les convertir en un nombre à virgule flottante double précision.

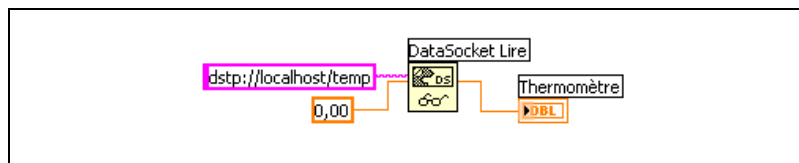


Figure 17-2. Lecture d'une valeur unique à l'aide de DataSocket Lire

Convertissez des données directement en un type particulier en câblant une commande ou une constante à l'entrée **type** de la fonction DataSocket Lire. Si vous ne spécifiez pas de type, la sortie **données** de la fonction DataSocket Lire renvoie des données de type variant, que vous devez alors manipuler à l'aide de la fonction Variant en donnée située dans la palette **Fonctions»Communication»DataSocket»Variant**. Dans certains cas, vous devez convertir les données de type variant en données LabVIEW.

DataSocket et données variant

Dans certains cas, lorsque vous souscrivez aux données à partir d'une autre application par exemple, le VI, ou une autre application qui lit les données par programmation, ne peut pas reconverter les données à leur type initial. Vous pouvez aussi vouloir ajouter un attribut aux données, comme un horodatage ou une mise en garde, ce que les types de données n'autorisent pas.

Dans ce cas, utilisez la fonction En variant située dans la palette **Fonctions»Communication»DataSocket»Variant** pour convertir en données variant, par programmation, les données que vous écrivez dans une connexion DataSocket. La figure 17-3 présente un diagramme qui effectue l'acquisition continue d'une mesure de température, convertit les données en données variant et ajoute un horodatage aux données en tant qu'attribut.

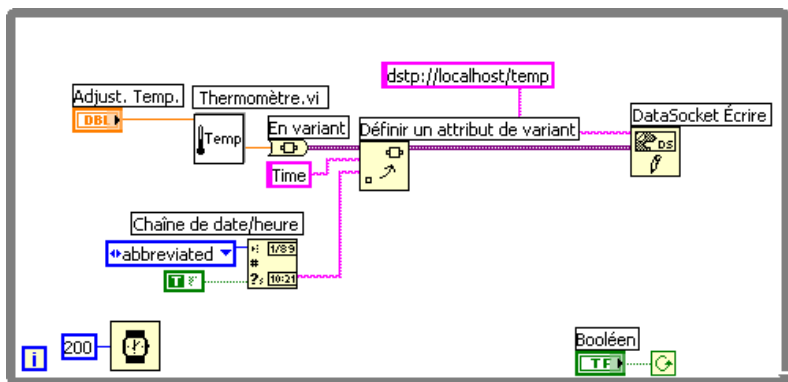


Figure 17-3. Conversion de données de température en direct en données variant

Quand un autre VI lit les données en direct, le VI doit convertir les données variant au type de données qu'il peut manipuler. La figure 17-4 présente un diagramme qui lit des données de température en continu à partir d'une connexion DataSocket, convertit les données variant en lecture de température, extrait l'attribut d'horodatage associé à chaque lecture et affiche la température et l'horodatage sur la face-avant.

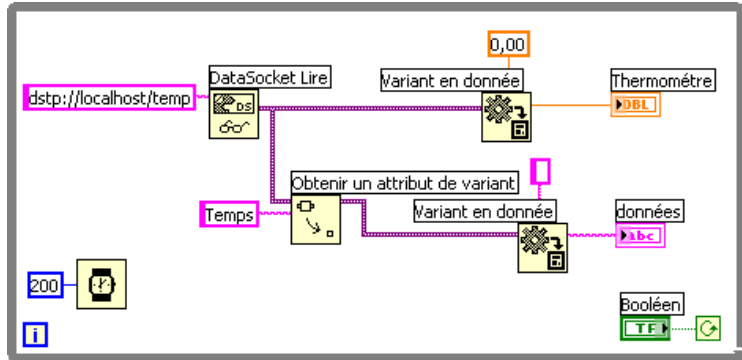


Figure 17-4. Conversion de données variant en direct en données de température

Publication de VIs sur le Web

Utilisez le serveur Web de LabVIEW pour créer des documents HTML et publier des images de faces-avant sur le Web. Vous pouvez contrôler l'accès du navigateur aux faces-avant publiées et définir quels VIs vont être visibles sur le Web.



Remarque Utilisez le LabVIEW Enterprise Connectivity Toolset pour contrôler des VIs sur le Web et pour ajouter des fonctions de sécurité supplémentaires aux VIs que vous publiez sur le Web. Allez sur le site de National Instruments à l'adresse ni.com si vous souhaitez obtenir plus d'informations sur ce toolset.

Options du serveur Web

Sélectionnez **Outils»Options...** et sélectionnez l'un des éléments du **Serveur Web** dans le menu déroulant du haut pour définir les options suivantes :

- Configurer le répertoire racine et le fichier journal.
- Activer le serveur Web.
- Contrôler l'accès du navigateur aux faces-avant du VI.
- Configurer quelles faces-avant de VI vont être visibles sur le Web.

Vous devez activer le serveur Web dans la page **Serveur Web : configuration** de la boîte de dialogue **Options** de LabVIEW avant de pouvoir publier des VIs sur le Web. Vous pouvez aussi activer le serveur Web au moyen de l'Outil de publication pour le Web, décrit à la section suivante. Les VIs doivent être en mémoire pour pouvoir être publiés.

Création de documents HTML

Sélectionnez **Outils»Outil de publication pour le Web...** pour utiliser l'outil de publication sur le Web et accomplir les tâches suivantes :

- Créer un document HTML.
- Insérer des images statiques ou animées de la face-avant dans un document HTML. Actuellement seuls les navigateurs Netscape acceptent les images animées.
- Insérer un VI que les clients peuvent visualiser et contrôler à distance.
- Ajouter du texte au-dessus et en dessous de l'image intégrée de la face-avant du VI.
- Placer un cadre autour d'une image ou d'un VI intégré.
- Visualiser un aperçu du document.
- Enregistrer le document sur disque.
- Activer, dans le serveur Web, la fonction de publication de documents HTML et d'images de face-avant sur le Web.

Publication d'images de faces-avant

Utilisez une URL `.snap` dans un navigateur Web ou un document HTML pour renvoyer une image statique de la face-avant d'un VI actuellement en mémoire. Les paramètres de requête de l'URL spécifient le nom du VI et les attributs de l'image. Par exemple, utilisez

`http://web.server.address/.snap?VIName.vi`, où `VIName.vi` est le nom du VI que vous voulez visualiser.

Utilisez une URL `.monitor` pour renvoyer une image animée de la face-avant d'un VI actuellement en mémoire. Les paramètres de requête de l'URL spécifient le nom du VI, les attributs de l'animation et les attributs de l'image. Par exemple, utilisez `http://web.server.address/.monitor?VIName.vi`, où `VIName.vi` est le nom du VI que vous voulez visualiser.

Formats des images de faces-avant

Le serveur Web peut générer des images des faces-avant au format d'image JPEG et PNG.

Le format JPEG compresse correctement les graphiques mais peut perdre certains détails graphiques. Ce format convient particulièrement bien aux photos. La compression JPEG des lignes, des faces-avant et des diagrammes peut provoquer un certain flou dans les graphiques et des

couleurs réparties inégalement. Le format PNG compresse également les graphiques correctement, mais parfois moins bien que le format JPEG. Cependant, la compression PNG n'entraîne aucune perte de qualité au niveau des détails.

Visualisation et contrôle des faces-avant à distance

Vous pouvez visualiser et contrôler une face-avant de VI à distance, soit à partir de LabVIEW, soit à partir d'un navigateur Web, en se connectant au serveur Web intégré de LabVIEW. Lorsque vous ouvrez une face-avant distante à partir de l'ordinateur d'un client, le serveur Web envoie la face-avant au client, mais le diagramme et tous les sous-VIs restent sur l'ordinateur du serveur. Vous pouvez interagir avec la face-avant de la même façon que si le VI s'exécutait sur le client, mis à part le fait que le diagramme s'exécute sur le serveur. Utilisez cette fonctionnalité pour publier des faces-avant entières ou pour contrôler vos applications à distance de manière sûre, aisée et rapide.



Remarque Utilisez le serveur Web de LabVIEW si vous voulez contrôler des VIs entiers. Utilisez le serveur DataSocket pour lire et écrire des données sur une commande unique de la face-avant d'un VI. Reportez-vous à la section [Utilisation de la technologie DataSocket](#) de ce chapitre pour plus d'informations sur l'utilisation du serveur DataSocket.

Configuration du serveur pour les clients

Depuis l'ordinateur du serveur, l'utilisateur doit d'abord configurer le serveur avant qu'un client ne puisse visualiser et contrôler une face-avant à distance en utilisant LabVIEW ou un navigateur Web. Configurez le serveur Web en sélectionnant **Outils»Options...** et les pages **Serveur Web** à partir du menu déroulant du haut. Utilisez ces pages pour contrôler l'accès au serveur et spécifier les faces-avant qui seront disponibles à distance. Vous pouvez également utiliser ces pages afin de fixer une limite de temps pour le contrôle d'un VI par un client distant particulier.

Le serveur Web permet à plusieurs clients de se connecter simultanément à la même face-avant, mais un seul client à la fois peut contrôler la face-avant. L'utilisateur peut, depuis l'ordinateur du serveur, reprendre le contrôle de n'importe quel VI à tout moment. Quand le contrôleur change une valeur sur la face-avant, toutes les faces-avant client reflètent ce changement. Cependant, les faces-avant client ne reflètent pas tous les changements. De manière générale, les faces-avant client ne reflètent pas les modifications apportées à l'affichage des objets de face-avant, mais bien celles apportées aux valeurs réelles de ces objets. Par exemple, si pour un

graphe déroulant, le contrôleur change le mode de représentation ou l'espacement des repères d'une échelle ou encore, si le contrôleur affiche et cache une barre de défilement, seule la face-avant du contrôleur reflétera ces modifications.

Licence pour face-avant distante

Vous devez configurer une licence pour supporter un nombre de clients qui pourraient potentiellement se connecter à votre serveur. Par défaut, la licence de face-avant distante comprise dans LabVIEW permet à un client unique de visualiser et de contrôler une face-avant. Lorsque vous excédez le nombre de clients autorisés par votre licence, le client reçoit un message qui contient des informations que vous spécifiez dans le fichier `LicenseErrorMessage.txt` situé dans le répertoire `labview\www`. Dans ce message, vous pouvez, si vous le souhaitez, inclure des informations concernant la personne que le client devrait contacter, comme par exemple l'administrateur du serveur ou la personne de votre société qui est responsable des mises à jour de la licence de face-avant distante. Si vous n'entrez rien dans ce fichier, un message d'erreur par défaut apparaîtra sur l'ordinateur du client pour indiquer que la demande de connexion a été refusée.



Remarque Bien que la version Système de développement complet ou la version Système de développement professionnel LabVIEW soient nécessaires pour construire des Vis que les clients pourront voir et contrôler à distance, n'importe quelle version de système de développement LabVIEW peut être utilisée pour voir et contrôler ces Vis à distance.

Visualisation et contrôle de faces-avant dans LabVIEW ou à partir d'un navigateur Web

Un client peut visualiser et contrôler une face-avant à distance à partir de LabVIEW ou d'un navigateur Web.



Remarque Avant de pouvoir visualiser et contrôler une face-avant à distance, vous devez activer le serveur Web sur l'ordinateur du serveur où se situe le VI que vous souhaitez visualiser.

Visualisation et contrôle des faces-avant dans LabVIEW

Pour visualiser une face-avant distante en utilisant LabVIEW comme client, ouvrez un nouveau VI et sélectionnez **Exécution»Connexion à la face-avant distante...** pour afficher la boîte de dialogue correspondante. Utilisez cette boîte de dialogue pour spécifier l'adresse Internet du serveur

et le VI que vous voulez visualiser. Par défaut, la face-avant de VI distante est initialement en mode observateur. Vous pouvez demander le contrôle en cochant l'option **Demander le contrôle** dans la boîte de dialogue **Connexion à une face-avant distante**. Quand le VI apparaît sur votre ordinateur, cliquez avec le bouton droit de la souris n'importe où sur la face-avant et sélectionnez **Demander le contrôle du VI** dans le menu local. Vous pouvez aussi accéder à ce menu en cliquant sur la barre d'état située en bas de la fenêtre de la face-avant. Si aucun autre client n'a actuellement le contrôle, un message apparaît pour indiquer que vous avez le contrôle de la face-avant. Si un autre client possède actuellement le contrôle du VI, le serveur garde votre demande dans la file d'attente jusqu'à ce que l'autre client abandonne le contrôle. Seul l'utilisateur situé à l'ordinateur du serveur peut surveiller la liste des clients dans la file d'attente en sélectionnant **Outils»Gestionnaire de connexions de face-avant distante**.

Tous les VIs que vous souhaitez pouvoir être visualisés et contrôlés par les clients doivent être en mémoire. Si le VI demandé est en mémoire, le serveur envoie la face-avant du VI au client demandeur. Si le VI n'est pas en mémoire, la section **État de la connexion** de la boîte de dialogue **Connexion à une face-avant distante** affiche un message d'erreur.

Visualisation et contrôle de faces-avant à partir d'un navigateur Web

Pour pouvoir visualiser ou contrôler une face-avant à distance via un navigateur Web, des clients n'ayant pas LabVIEW installé sur leur machine doivent installer le moteur d'exécution de LabVIEW. Le moteur d'exécution de LabVIEW inclut un module "plug-in" LabVIEW de navigation qui s'installe dans le répertoire plug-in du navigateur Web. Le CD de LabVIEW contient un installateur pour le moteur d'exécution de LabVIEW.

Les clients installent le moteur d'exécution de LabVIEW et l'utilisateur crée à l'ordinateur du serveur un fichier HTML comprenant un tag <OBJECT> qui référence le VI que vous souhaitez rendre visible et contrôlable par les clients. Ce tag contient une référence URL vers un VI et des informations qui guide le navigateur Web pour transmettre le VI au module plug-in de navigation de LabVIEW. Les clients naviguent vers le serveur Web en entrant l'adresse de celui-ci dans le champ adresse ou URL situé tout en haut de la fenêtre du navigateur Web. Le module plug-in affiche la face-avant dans la fenêtre du navigateur Web et communique avec

le serveur Web de façon à ce que le client puisse interagir avec la face-avant distante. Les clients demandent le contrôle en sélectionnant **Demander le contrôle du VI** en bas de la fenêtre de la face-avant distante dans leur navigateur Web ou en cliquant n'importe où sur la face-avant avec le bouton droit de la souris et en sélectionnant **Demander le contrôle du VI** dans le menu local.



Remarque National Instruments vous conseille d'utiliser Netscape 4.7 ou version ultérieure, ou encore Internet Explorer 5.0 ou version ultérieure pour visualiser et contrôler des faces-avant distantes à l'aide d'un navigateur Web.

Fonctionnalité non supportée dans la visualisation et le contrôle des faces-avant distantes

Du fait des contraintes imposées par un navigateur Web, les applications de type interface utilisateur qui essaient de manipuler les dimensions et la position d'une face-avant ne fonctionnent pas correctement lorsque la face-avant est affichée en tant que partie d'une page Web. Bien que le serveur Web et le module plug-in de navigation de LabVIEW essaient de rester fidèles aux applications interface utilisateur complexes (en particulier celles qui présentent des boîtes de dialogue et des fenêtres de sous-VIs), certaines applications peuvent ne pas fonctionner correctement dans le contexte d'un navigateur Web. National Instruments vous recommande de ne pas exporter ces types d'applications pour les utiliser dans un navigateur Web.

En général, évitez d'exporter des VIs faisant intervenir un nombre très élevé de données pour la visualisation et le contrôle à distance. Par exemple, des faces-avant distantes possédant plusieurs graphes déroulants augmentent la charge réseau, ce qui entraîne un ralentissement des mises à jour de face-avant. De même, les VIs possédant des boucles While sans fonction d'attente peuvent empêcher la réalisation des tâches de second plan dans un temps raisonnable, entraînant ainsi un manque de réponse des faces-avant lorsqu'elles sont visualisées ou contrôlées à distance.

Par ailleurs, certains VIs peuvent ne pas fonctionner exactement de la même façon à partir d'un ordinateur distant que lorsqu'ils s'exécutent localement. Les contrôles ActiveX intégrés dans une face-avant ne s'affichent pas sur un client distant car ils fonctionnent de façon presque totalement indépendante de LabVIEW. Si un VI présente la boîte de dialogue de fichiers standard, le contrôleur reçoit une erreur car vous ne pouvez pas naviguer à distance dans un système de fichiers. Le bouton **Parcourir** d'une commande chemin est également désactivé dans des faces-avant distantes.

Il se peut que des clients visualisant une face-avant à distance observent un comportement différent si la face-avant à laquelle ils se connectent provient d'un exécutable. En particulier, si la face-avant provient d'un exécutable, tous les changements effectués par programmation à la face-avant avant que le client ne se connecte à celle-ci ne sont pas reflétés sur l'ordinateur du client. Par exemple, si un nœud de propriété modifie le sous-titre d'une commande avant qu'un client ne se connecte à la face-avant, le client verra le sous-titre original de la commande et non pas le sous-titre modifié.

Il se peut que des diagrammes qui arrivent à obtenir certains effets de type interface utilisateur grâce aux propriétés d'une commande de face-avant rencontrent une diminution de performances lorsque vous contrôlez le VI à partir d'un ordinateur distant. Vous pouvez améliorer les performances de ces VIs en utilisant la fonction Attendre une activité sur la face-avant.

Applications de communication de bas niveau

LabVIEW supporte plusieurs protocoles de bas niveau que vous pouvez utiliser pour communiquer d'un ordinateur à un autre.

Les protocoles diffèrent tous les uns des autres, en particulier dans la manière dont ils font référence à l'endroit du réseau où se trouve une application à distance. Les protocoles sont généralement incompatibles entre eux. Par exemple, pour établir une communication entre Macintosh et Windows, vous devez utiliser un protocole qui fonctionne sur les deux plates-formes, TCP par exemple.

TCP et UDP

Transmission Control Protocol (TCP) et User Datagram Protocol (UDP) sont disponibles sur toutes les plates-formes que LabVIEW supporte. Le protocole TCP est un protocole de connexion fiable. Il peut détecter les erreurs et il veille à ce que les données arrivent en ordre et sans duplication. Ces qualités font du protocole TCP le protocole le mieux adapté aux applications réseau.

Bien que le protocole UDP soit plus performant que le protocole TCP et n'ait pas besoin de connexion, il ne garantit pas la livraison. Utilisez le protocole UDP dans des applications dans lesquelles la livraison n'est pas d'une importance capitale. Par exemple, une application peut transmettre des données à une destination suivant une fréquence suffisante pour que la perte de quelques segments de données ne pose pas de problème. Reportez-vous à la note d'application *Utilisation de LabVIEW avec TCP/IP*

et *UDP* pour obtenir de plus amples informations sur l'utilisation des protocoles TCP et UDP dans les applications de LabVIEW.

DDE (Windows)

Dynamic Data Exchange (DDE) fonctionne à un niveau plus élevé que TCP pour échanger des commandes et des données entre les clients et les serveurs. Le DDE est bien adapté à la communication avec les applications standard du commerce comme Microsoft Excel. Reportez-vous à la note d'application *Using DDE in LabVIEW* pour obtenir de plus amples informations sur l'utilisation du protocole DDE dans les applications de LabVIEW.

Apple Events et PPC Toolbox (Macintosh)

La forme de communication réservée à Macintosh la plus courante est Apple Events. Comme avec le protocole DDE, utilisez Apple Events pour envoyer des messages de demande d'intervention ou de retour d'informations en provenance d'autres applications de Macintosh.

La boîte à outils Communication de programme à programme (PPC Toolbox) est un moyen fonctionnant à plus bas niveau pour envoyer et recevoir des données entre applications de Macintosh. La boîte à outils PPC est plus performante que Apple Events parce qu'elle utilise moins de temps système pour transmettre des informations. Cependant, la boîte à outils PPC ne définissant pas les types d'informations que vous pouvez transférer, de nombreuses applications ne l'acceptent pas. La boîte à outils PPC est la méthode la plus performante d'envoi de grandes quantités d'informations entre applications l'acceptant. Reportez-vous à la note d'application *Using Apple Events and the PPC Toolbox to Communicate with LabVIEW Applications on the Macintosh* pour de plus amples informations sur l'utilisation d'Apple Events et de la PPC Toolbox dans les applications de LabVIEW.

Vis de canaux (UNIX)

Utilisez les VIs de canaux de communication pour ouvrir, fermer, lire et écrire dans des canaux de communication UNIX. Utilisez les canaux de communication pour communiquer entre LabVIEW et des processus avec lesquels il n'y a pas de lien.

Exécution des commandes système (Windows et UNIX)

Au moyen du VI System Exec, vous exécutez une ligne de commande au niveau du système qui peut inclure tous les paramètres supportés par l'application que vous voulez lancer.

ActiveX

Avec ActiveX Automation, une application Windows telle que LabVIEW fournit un ensemble d'objets, de commandes et de fonctions auxquelles peuvent accéder d'autres applications de Windows. Vous pouvez utiliser LabVIEW comme client ActiveX pour accéder aux objets, aux propriétés, aux méthodes et aux événements associés à d'autres applications pour lesquelles ActiveX est activé. LabVIEW peut aussi servir de serveur ActiveX, ce qui signifie que d'autres applications pour lesquelles ActiveX est activé peuvent accéder à des objets, des propriétés et des méthodes de LabVIEW.

Dans ce manuel, *ActiveX* se rapporte aux technologies OLE et ActiveX de Microsoft. Cette technologie n'est disponible que sur Windows. Reportez-vous à la documentation de Microsoft Developer's Network, *Inside OLE*, de Kraig Brockschmidt, seconde édition, et à *Essential COM*, de Don Box, pour obtenir de plus amples informations sur ActiveX.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur l'utilisation de la technologie ActiveX.

Objets, propriétés, méthodes et événements ActiveX

Les applications pour lesquelles ActiveX est activé incluent les objets dont les propriétés et les méthodes exposées sont accessibles à d'autres applications. Les objets peuvent être des objets visibles par les utilisateurs, comme des boutons, des fenêtres, des images, des documents et des boîtes de dialogue ou des objets que l'utilisateur ne peut pas visualiser, comme des objets d'applications. Vous accédez à une application en accédant à un objet associé à cette application, en paramétrant une propriété ou en appelant une méthode de cet objet.

Reportez-vous à la section *Édition des paramètres de l'application et du VI* du chapitre 16, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur les objets, les propriétés et les méthodes.

Les événements sont des actions effectuées sur un objet, tel que cliquer sur une souris, appuyer sur une touche ou manquer de mémoire. Chaque fois que ces actions s'effectuent sur l'objet, celui-ci envoie un événement pour prévenir le conteneur ActiveX en même temps que les données propres à l'événement.

Reportez-vous au VI ActiveX Event Template et au VI List ActiveX Events dans `examples\comm\axevent.llb`, ainsi qu'au VI Family Tree et au VI MultiEvents du répertoire `examples\comm` si vous voulez consulter des exemples d'utilisation d'événements ActiveX dans LabVIEW et des exemples d'utilisation des VIs ActiveX Events.

VIs, fonctions, commandes et indicateurs ActiveX

Utilisez les VIs, les fonctions, les commandes et les indicateurs suivants pour accéder aux objets, aux propriétés, aux méthodes et aux événements associés à d'autres applications pour lesquelles ActiveX est activé :

- Utilisez la commande `refnum automation`, qui se trouve dans la palette **Commandes»ActiveX**, pour créer une référence à un objet ActiveX. Cliquez sur cette commande de la face-avant avec le bouton droit de la souris pour sélectionner un objet de la bibliothèque de type à laquelle vous voulez accéder.
- Utilisez la fonction `Automation Ouvrir` située dans la palette **Fonctions»Communication»ActiveX** pour ouvrir un objet ActiveX.
- Utilisez le conteneur ActiveX, qui se trouve dans la palette **Commandes»ActiveX**, pour accéder à un objet ActiveX et l'afficher sur la face-avant. Cliquez sur cette commande avec le bouton droit de la souris pour sélectionner l'objet auquel vous voulez accéder.
- Utilisez le nœud de propriété, qui se trouve dans la palette **Fonctions»Communication»ActiveX**, pour obtenir (lire) et définir (écrire) les propriétés associées à un objet ActiveX.
- Utilisez le nœud de méthode, qui se trouve dans la palette **Fonctions»Communication»ActiveX**, pour appeler les méthodes associées à un objet ActiveX.
- Utilisez les VIs d'événements ActiveX, qui se trouvent dans la palette **Fonctions»Communication»ActiveX»Événements ActiveX**, pour gérer les événements qui se produisent sur un objet ActiveX que vous avez placé dans le conteneur ActiveX de la face-avant.
- Utilisez la commande et l'indicateur variant ActiveX, qui se trouvent dans la palette **Commandes»ActiveX**, pour transmettre des données vers ou provenant de sous-VIs. Vous pouvez aussi utiliser cette commande et cet indicateur pour afficher les données sans les modifier.

Reportez-vous à la section *Manipulation de données variant* du chapitre 5, *Construction du diagramme*, pour obtenir plus de détails sur les données variant.

LabVIEW en tant que client ActiveX

Lorsque LabVIEW accède à des objets associés à une autre application pour laquelle ActiveX est activé, il fait office de client ActiveX. Vous pouvez utiliser LabVIEW comme client ActiveX de l'une des manières suivantes :

- Ouvrir une application (Microsoft Excel, par exemple), créer un document et lui ajouter des données.
- Intégrer un document, tel qu'un document Microsoft Word ou un document de type tableur d'Excel, sur la face-avant.
- Placer un bouton ou tout autre objet d'une autre application sur la face-avant, par exemple un bouton d'**Aide** qui appelle le fichier d'aide de l'autre application.
- Lier une commande ActiveX que vous avez construite dans une autre application.

LabVIEW accède à un objet ActiveX par la commande refnum Automation et par le conteneur ActiveX qui sont tous deux des objets de la face-avant. Utilisez la commande refnum Automation pour sélectionner un objet ActiveX. Utilisez le conteneur ActiveX pour sélectionner un objet ActiveX affichable, comme un bouton ou un document, et le placer sur la face-avant. Les objets s'affichent tous deux comme commandes refnums automation dans le diagramme.


Accès à une application pour laquelle ActiveX est activé

Pour accéder à une application pour laquelle ActiveX est activé, utilisez la commande refnum automation du diagramme pour créer une référence à une application. Câblez la commande refnum automation à la fonction Automation Ouvrir, qui ouvre l'application d'appel. Utilisez le nœud de propriété pour sélectionner les propriétés associées à l'objet et y accéder. Utilisez le nœud de méthode pour sélectionner les méthodes associées à l'objet et y accéder. Fermez la référence à l'objet au moyen de la fonction Automation Fermer. Lorsque vous fermez la référence, l'objet est supprimé de la mémoire.

Par exemple, vous pouvez construire un VI qui ouvre Microsoft Excel pour qu'il s'affiche sur l'écran de l'utilisateur, crée un classeur, un fichier de type tableur, une table dans LabVIEW et écrive cette table dans le fichier de type tableur d'Excel.

Reportez-vous au VI Write Table dans XL dans `examples\comm\ExcelExamples.llb` si vous voulez consulter un exemple d'utilisation de LabVIEW en tant que client Excel.



Remarque Les applications qui comprennent des interfaces ActiveX personnalisées apparaissent avec une icône . Cliquez sur l'icône afin de sélectionner un objet pour l'interface personnalisée. Reportez-vous à la section [Support pour les interfaces ActiveX Automatisation personnalisées](#) de ce chapitre pour obtenir plus d'informations sur les interfaces personnalisées.

Insertion d'un objet ActiveX sur la face-avant

Pour insérer une commande ActiveX sur la face-avant, utilisez le conteneur ActiveX afin de sélectionner l'objet ou le document ActiveX et accéder aux propriétés et aux méthodes associées à cet objet. Vous pouvez définir les propriétés d'un objet ActiveX en utilisant le navigateur de propriétés ActiveX ou les pages de propriétés. Vous pouvez également définir les propriétés par programmation à l'aide du nœud de propriété. Reportez-vous à la section [Définition des propriétés ActiveX](#) de ce chapitre pour obtenir plus de détails sur la définition des propriétés ActiveX.

Utilisez le nœud de méthode pour appeler les méthodes associées à l'objet.

Par exemple, vous pouvez afficher une page Web sur la face-avant au moyen du conteneur ActiveX pour accéder à un objet du navigateur Web de Microsoft, en sélectionnant la classe de méthodes Naviguer, puis la méthode URL et en spécifiant l'URL.

Si vous utilisez le conteneur ActiveX, vous n'avez pas à câbler la commande refnum automation du diagramme à la fonction Automatisation Ouvrir ni à fermer la référence à l'objet au moyen de la fonction Automatisation Fermer. Vous pouvez câbler directement le nœud de propriété ou le nœud de méthode car le conteneur ActiveX intègre l'application appelante dans LabVIEW. Cependant, si le conteneur ActiveX inclut des propriétés et des méthodes qui renvoient d'autres refnums automation, vous devez fermer ces autres commandes refnums.

Définition des propriétés ActiveX

Après avoir appelé ou inséré un objet ou document ActiveX, vous pouvez définir les propriétés qui lui sont associées en utilisant le navigateur de propriétés ActiveX, les pages de propriétés et le nœud de propriété.

Navigateur de propriétés ActiveX

Utilisez le navigateur de propriétés ActiveX pour visualiser et définir toutes les propriétés associées à un objet ou document ActiveX compris dans un conteneur ActiveX. Pour accéder au navigateur de propriétés ActiveX, cliquez avec le bouton droit de la souris sur l'objet ou le document situé dans le conteneur de la face-avant et sélectionnez **Navigateur de propriétés...** dans le menu local. Vous pouvez également sélectionner **Outils»Avancé»Navigateur de propriétés ActiveX....** Le navigateur de propriétés ActiveX représente un moyen aisé de définir les propriétés d'un contrôle ActiveX de manière interactive, mais vous ne pouvez pas l'utiliser pour définir les propriétés par programmation. Par ailleurs, vous ne pouvez utiliser le navigateur de propriétés ActiveX qu'avec des objets situés dans un conteneur. Le navigateur de propriétés ActiveX n'est pas disponible en mode Exécution.

Pages de propriétés ActiveX

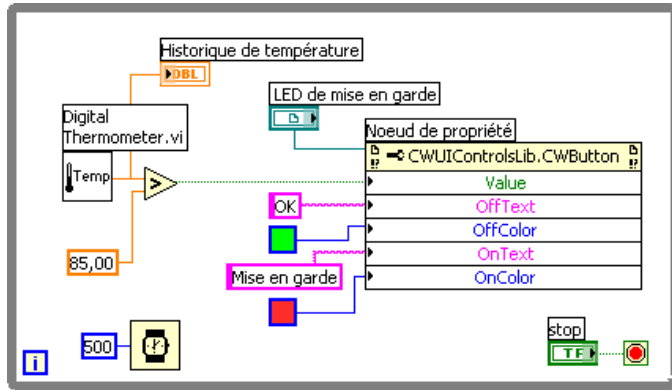
Beaucoup d'objets ActiveX comprennent des pages de propriétés, qui organisent les propriétés associées aux objets sur des onglets séparés. Pour accéder aux pages de propriétés ActiveX, cliquez avec le bouton droit de la souris sur l'objet ou le document situé dans le conteneur de la face-avant et sélectionnez le nom de l'objet dans le menu local.

Tout comme le navigateur de propriétés ActiveX, les pages de propriétés ActiveX représentent un moyen aisé de définir les propriétés d'un contrôle ActiveX de manière interactive, mais vous ne pouvez pas les utiliser pour définir les propriétés par programmation. Par ailleurs, vous ne pouvez utiliser les pages de propriétés qu'avec des objets ActiveX situés dans un conteneur. De plus, les pages de propriétés ActiveX ne sont pas disponibles pour tous les objets ActiveX.

Nœuds de propriété

Utilisez le nœud de propriété pour définir les propriétés ActiveX par programmation. Par exemple, si vous utilisez un contrôle ActiveX pour avertir un utilisateur quand une température dépasse une certaine limite, utilisez le nœud de propriété et définissez la propriété de valeur de l'objet afin de spécifier la limite.

L'exemple suivant modifie la propriété Value de l'objet ActiveX CWButton, qui fait partie de la bibliothèque de commandes de Component Works, quand la température atteint ou dépasse 85° Fahrenheit.



Dans ce cas, l'objet CWButton se comporte comme un voyant (LED), change de couleur et affiche `Alerte` quand la température atteint la limite, ce qui correspond à l'état ON de l'objet CWButton.



Remarque Dans cet exemple, vous auriez pu utiliser le navigateur de propriétés ActiveX ou les pages de propriétés pour définir les propriétés `OffText`, `OffColor`, `OnText` et `OnColor` pour l'objet CWButton car vous n'avez pas besoin de définir ces propriétés par programmation. Reportez-vous aux sections [Navigateur de propriétés ActiveX](#) et [Pages de propriétés ActiveX](#) de ce chapitre pour obtenir plus d'informations sur respectivement le navigateur de propriétés ActiveX et les pages de propriétés.

LabVIEW en tant que serveur ActiveX

Les applications, les VIs, les propriétés et les méthodes de commande de LabVIEW sont accessibles par des appels ActiveX à partir d'autres applications. D'autres applications pour lesquelles ActiveX est activé, comme Microsoft Excel, peuvent demander des propriétés, des méthodes et des VIs individuels à LabVIEW qui va alors faire office de serveur ActiveX.

Par exemple, vous pouvez intégrer un VI graphe dans un fichier au format tableur d'Excel et, à partir du fichier au format tableur, entrer des données dans les entrées du VI et exécuter le VI. Lorsque vous exécutez le VI, les données s'enregistrent sous forme de courbe dans le graphe.

Reportez-vous à `exemples\comm\freqresp.xls` si vous souhaitez consulter un exemple d'utilisation des propriétés et des méthodes de LabVIEW dans un fichier au format tableur d'Excel.

Support pour les interfaces ActiveX Automation personnalisées

En tant que client ActiveX qui accède aux propriétés et méthodes provenant d'un serveur ActiveX en utilisant LabVIEW, vous pouvez accéder aux interfaces personnalisées exposées par le serveur. Celui qui écrit le serveur ActiveX doit s'assurer que les paramètres des propriétés et méthodes contenues dans ces interfaces personnalisées sont des données de type Automation (IDispatch). Reportez-vous à la documentation sur l'environnement de programmation de votre serveur pour plus d'informations concernant l'accès aux interfaces personnalisées.

Utilisation de constantes pour définir des paramètres dans les VIs ActiveX

Certains paramètres des nœuds ActiveX dressent une liste d'un certain nombre de valeurs valides. Sélectionnez le nom de description dans la constante menu déroulant pour définir ces valeurs de paramètres. Pour accéder à la constante menu déroulant pendant la construction d'un VI ActiveX, cliquez sur le paramètre du nœud acceptant les valeurs des données avec le bouton droit de la souris et sélectionnez **Créer une constante** dans le menu local. Les sélections disponibles dans la constante menu déroulant sont fonction des refnums acheminés au nœud. Les figures 18-1 et 18-2 présentent des exemples d'utilisation de la constante menu déroulant et de la constante numérique pour définir les valeurs des paramètres.

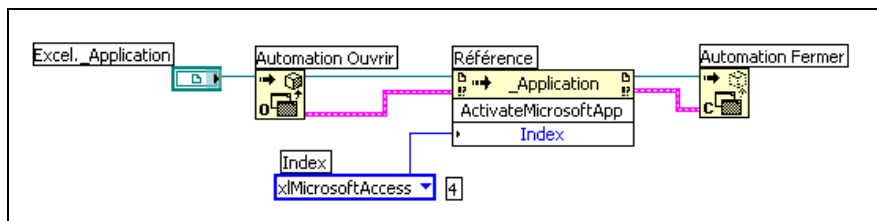


Figure 18-1. Paramétrage d'une valeur de donnée à l'aide d'une constante menu déroulant

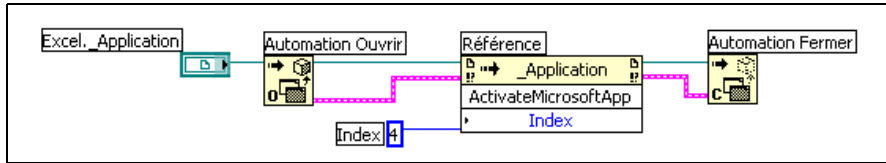


Figure 18-2. Paramétrage d'une valeur de donnée à l'aide d'une constante numérique

Les paramètres acceptant des valeurs de données ont une petite flèche à gauche de leur nom. Pour visualiser la valeur des données numériques correspondante, cliquez sur la constante menu déroulant avec le bouton droit de la souris et sélectionnez **Éléments visibles»Afficheur numérique** dans le menu local.

Les VIs des figures 18-1 et 18-2 accèdent tous deux à l'application Microsoft Excel et exécutent une méthode. Le paramètre **Indice** de la méthode **ActivateMicrosoftApp** comporte les options suivantes : **MicrosoftWord**, **MicrosoftPowerPoint**, **MicrosoftMail**, **MicrosoftAccess**, **MicrosoftFoxPro**, **MicrosoftProject** et **MicrosoftSchedulePlus**.

Pour identifier la valeur numérique du paramètre **Indice** correspondant à l'option **MicrosoftAccess** dans la figure 18-1, sélectionnez l'option **MicrosoftAccess** dans le menu déroulant. La valeur numérique de l'option sélectionnée actuelle s'affiche dans une boîte à côté de la constante menu déroulant. Au lieu d'utiliser une constante menu déroulant, vous pouvez entrer la valeur numérique d'une option dans une constante numérique, comme l'illustre la figure 18-2.

Appel de code provenant de langages de programmation textuels

Vous pouvez appeler la plupart des bibliothèques partagées standard de LabVIEW en utilisant la fonction Appeler une fonction d'une DLL. Vous pouvez aussi appeler du code C dans LabVIEW au moyen du Code Interface Node ou CIN.

Reportez-vous au manuel *Using External Code in LabVIEW* pour obtenir de plus amples informations sur l'appel de codes de langages de programmation textuels.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur l'appel des codes de langages de programmation textuels.

Appeler une fonction d'une DLL

Utilisez la fonction Appeler une fonction d'une DLL pour appeler la plupart des bibliothèques partagées standard ou DLL. Cette fonction vous permet de créer une interface dans LabVIEW qui va servir à appeler des bibliothèques existantes ou de nouvelles bibliothèques écrites tout spécialement pour être utilisées avec LabVIEW. National Instruments recommande l'utilisation de la fonction Appeler une fonction d'une DLL pour créer une interface vers un code externe.

Code Interface Node

Utilisez le CIN comme une méthode alternative d'appel de code source écrit en langage C. La fonction Appeler une fonction d'une DLL est généralement plus facile à utiliser que le CIN.

Formules et équations

Pour utiliser une équation complexe dans LabVIEW, il n'est pas nécessaire d'assembler des fonctions arithmétiques variées dans le diagramme. Vous pouvez développer des équations dans un environnement mathématique familier et les intégrer ensuite dans une application.

Utilisez la boîte de calcul pour effectuer des opérations mathématiques à l'intérieur de l'environnement LabVIEW. Pour accéder à des fonctions plus avancées, vous pouvez utiliser les applications mathématiques de HiQ et MATLAB pour développer des équations. HiQ et MATLAB sont des logiciels qui vous aident à organiser et à visualiser des problèmes réels de type mathématique, scientifique et technique.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur l'utilisation des équations et de la syntaxe à employer, sur les fonctions et opérateurs existants, et sur les descriptions d'erreurs éventuelles.

Méthodes d'utilisation d'équations dans LabVIEW

Vous pouvez utiliser la boîte de calcul, le nœud du script HiQ et le nœud de script MATLAB pour effectuer des opérations mathématiques dans le diagramme.



Remarque HiQ ou MATLAB doit être installé sur votre ordinateur pour utiliser les nœuds de script, car LabVIEW utilise la technologie ActiveX pour faire transférer le script à exécuter dans HiQ ou dans MATLAB. LabVIEW utilise la technologie ActiveX, disponible uniquement sous Windows, pour réaliser les nœuds de script. Par conséquent, les nœuds de script ne sont disponibles que sous Windows.

National Instruments fournissant HiQ avec LabVIEW, vous pouvez installer le logiciel sans frais supplémentaire pour augmenter le nombre de fonctions de gestion des équations.

Les nœuds de script sont similaires à la boîte de calcul, mais ils vous permettent d'importer un script HiQ ou un script MATLAB existant en code ASCII et d'exécuter le script dans LabVIEW. Comme avec une boîte de calcul, vous pouvez transmettre des données au nœud et à partir du nœud.

Boîtes de calcul

La boîte de calcul est un nœud textuel pratique que vous pouvez utiliser pour effectuer des opérations mathématiques dans le diagramme. Vous n'avez pas à accéder à un code externe ni à des applications, et vous n'avez pas à câbler de fonctions arithmétiques de bas niveau pour créer des équations. En plus des expressions d'équations textuelles, la boîte de calcul peut accepter les versions textuelles de déclarations If, de boucles While, de boucles For et de boucles Do, qui sont des structures bien connues des programmeurs en langage C. Ces éléments de programmation et les éléments de programmation en langage C se ressemblent mais ne sont toutefois pas identiques.

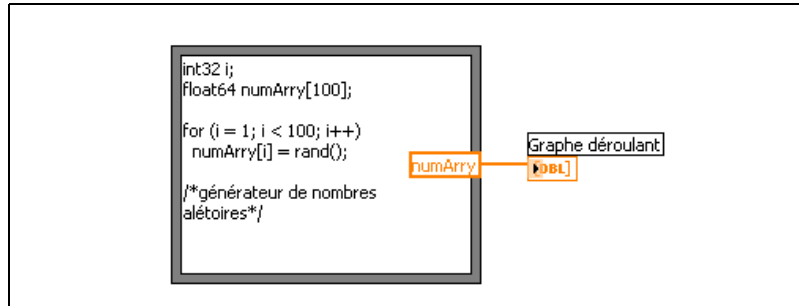
Les boîtes de calcul sont utiles pour résoudre des équations compliquées, comme les équations qui comportent de nombreuses variables, et pour utiliser un code textuel existant. Vous pouvez copier et coller le code textuel existant dans une boîte de calcul au lieu de le recréer graphiquement.

Les boîtes de calcul procèdent à une vérification de type pour vérifier que les indices de tableau sont bien des données numériques et que les opérandes associées aux opérations bit à bit sont des entiers. Les boîtes de calcul vérifient aussi que les indices de tableau sont compris dans la bonne gamme. Pour les tableaux, une valeur hors gamme prend, par défaut, la valeur zéro et une opération hors gamme indique `NOP` pour montrer qu'aucune opération n'a eu lieu.

Les boîtes de calcul exécutent également des conversions de type automatique.

Utilisation de la boîte de calcul

La boîte de calcul située dans les palettes **Fonctions»Structures** et **Fonctions»Mathématiques»Formule** est une boîte redimensionnable similaire à la boucle For, à la boucle While, à la structure Condition et à la structure Séquence. Cependant, à la place d'un sous-diagramme, la boîte de calcul contient au moins une instruction en langage C, délimitée par des points-virgules, comme illustré dans l'exemple ci-après. Comme en langage C, ajoutez des commentaires en les entourant de deux paires barre oblique/astérisque (`/*commentaire*/`).



Reportez-vous au VI Equations dans `exemples\general\structs.llb` si vous voulez consulter un exemple d'utilisation de boîte de calcul.

Variables dans la boîte de calcul

Lorsque vous travaillez avec des variables, gardez les points suivants à l'esprit :

- Il n'y a pas de limite au nombre de variables ou d'équations dans la boîte de calcul.
- Deux entrées et deux sorties ne peuvent pas porter le même nom, mais une sortie peut avoir le même nom qu'une entrée.
- Déclarez une variable d'entrée en cliquant sur le bord de la boîte de calcul avec le bouton droit de la souris et en sélectionnant **Ajouter une entrée** dans le menu local. Vous ne pouvez pas déclarer de variables d'entrée à l'intérieur de la boîte de calcul.
- Déclarez une variable de sortie en cliquant sur le bord de la boîte de calcul avec le bouton droit de la souris et en sélectionnant **Ajouter une sortie** dans le menu local. Le nom de la variable de sortie doit correspondre soit au nom d'une variable d'entrée, soit au nom d'une variable que vous déclarez à l'intérieur de la boîte de calcul.
- Vous pouvez changer une variable d'entrée en variable de sortie et vice versa, en sélectionnant **Changer en entrée** ou **Changer en sortie** dans le menu local.
- Vous pouvez déclarer et utiliser une variable dans la boîte de calcul sans la relier à un fil de liaison d'entrée ou de sortie.
- Vous devez câbler tous les terminaux d'entrée.

- Les variables peuvent être des scalaires de type flottant dont la précision varie en fonction de la configuration de votre ordinateur. Vous pouvez aussi utiliser des entiers et des tableaux de nombres comme variables.
- Les variables ne peuvent pas avoir d'unités.

Nœuds d'expression

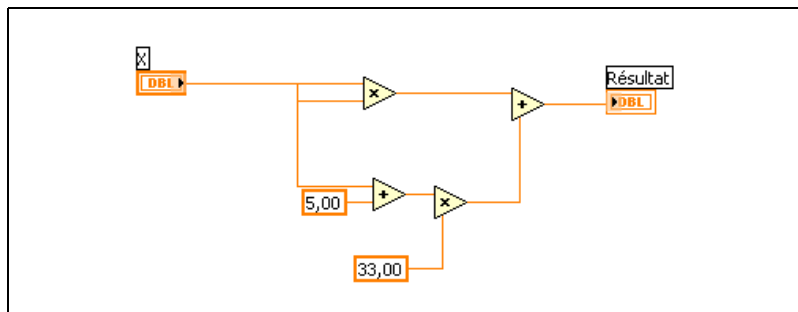
Utilisez les nœuds d'expression pour calculer des expressions ou des équations contenant une seule variable. Les nœuds d'expression sont utiles quand une équation n'a qu'une seule variable mais qui est, cependant, compliquée.

Les nœuds d'expression utilisent la valeur que vous acheminez au terminal d'entrée comme valeur de la variable. Le terminal de sortie renvoie la valeur du calcul.

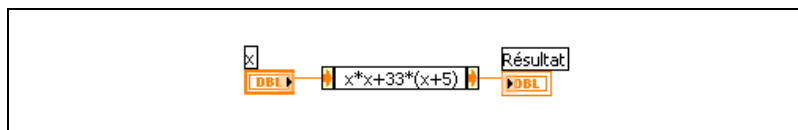
Par exemple, considérons l'équation simple suivante :

$$x \times x + 33 \times (x + 5)$$

Le diagramme de la figure suivante utilise des fonctions numériques pour représenter cette équation.



Utilisez un nœud d'expression, comme illustré dans la figure suivante, pour créer un diagramme beaucoup plus simple.



Polymorphisme dans les nœuds d'expression

Le terminal d'entrée d'un nœud d'expression est du même type de données que la commande ou la constante à laquelle vous le câblez. Le terminal de sortie est du même type de données que le terminal d'entrée. Le type de données de l'entrée peut être tout nombre scalaire non complexe, tout tableau de nombres scalaires non complexes ou un cluster de nombres scalaires non complexes. En ce qui concerne les tableaux et les clusters, le nœud d'expression applique l'équation à chaque élément d'un tableau ou d'un cluster d'entrée.

Utilisation de HiQ avec LabVIEW

HiQ est un environnement haute performance de résolution interactive de problèmes dans lequel vous pouvez analyser, visualiser et documenter des problèmes scientifiques et techniques du monde réel. HiQ utilise une interface notebook virtuelle et un langage de programmation appelé Script HiQ. Le notebook est divisé en pages, en sections et en onglets dans lesquels vous pouvez placer et organiser des objets, tels que du texte, des nombres et des graphes.

Utilisez les VIs situés dans la palette **Fonctions»Communication»HiQ** pour contrôler HiQ à partir de LabVIEW et transférer des données d'une application à l'autre, dans des formats divers. Vous devez installer HiQ afin d'utiliser ces VIs. Utilisez ces VIs HiQ pour lancer HiQ, accéder aux notebooks HiQ et les manipuler afin d'analyser et visualiser les données, et exécuter un script HiQ à l'intérieur d'un notebook.

Reportez-vous à l'*Aide HiQ* pour obtenir de plus amples informations sur les fonctions HiQ, y compris les options mathématiques, la référence à la fonction et les informations de la syntaxe.

Reportez-vous à `exemples\comm\hiq` pour consulter des exemples de configuration et d'utilisation des VIs HiQ.

Nœuds de script HiQ et MATLAB

Utilisez les nœuds de script HiQ et MATLAB situés dans la palette **Fonctions»Mathématiques»Formule** pour charger et éditer les scripts HiQ et MATLAB dans le diagramme pour que LabVIEW puisse travailler avec leurs fonctions mathématiques avancées.

HiQ ou MATLAB doit être installé pour que vous puissiez utiliser les nœuds de script. Si vous avez déjà un script écrit en HiQ ou en MATLAB, vous pouvez l'importer dans le nœud de script.

Vous définissez les terminaux de nœuds de script comme entrées ou sorties pour les variables du script, afin d'acheminer les valeurs de HiQ ou de MATLAB à LabVIEW et vice versa. Vous pouvez déterminer la fonction d'un terminal en fonction de la manière dont l'équation est écrite. Ainsi, si le script contient la déclaration d'affectation, $X = i + 3$, vous pouvez définir i comme terminal d'entrée pour contrôler de quelle façon le nœud de script va calculer X , et vous pouvez affecter X à un terminal de sortie pour extraire le résultat final du calcul du script.

Si vous n'avez pas encore de script écrit, vous pouvez placer un nœud de script dans le diagramme et créer un script au moyen de la syntaxe HiQ ou de la syntaxe MATLAB. LabVIEW communique avec le moteur du serveur de script, qui est un programme qui exécute le script. LabVIEW communique et contrôle le moteur du serveur de script au moyen d'un protocole industriel. HiQ et MATLAB sont installés dans les moteurs de serveur de script.

En raison de la nature des langages de scripts HiQ et MATLAB, le nœud de script ne peut pas déterminer le type de données des terminaux que vous avez créés. Vous devez assigner un type de données LabVIEW à chaque terminal de nœud de script. Le nœud de script dans LabVIEW reconnaît les types de données acceptés par HiQ et MATLAB. La table 20-1 illustre les types de données LabVIEW et les types de données correspondants dans HiQ et MATLAB.

Tableau 20-1. Types de données LabVIEW, HiQ et MATLAB











LabVIEW Type de données	HiQ Type de données	MATLAB Type de données
	Entier	—
	Réel	Réel
	Texte	—
	Vecteur entier	—
	Vecteur réel	Vecteur réel
	Matrice entière	—
	Matrice réelle	Matrice réelle

Tableau 20-1. Types de données LabVIEW, HiQ et MATLAB (Suite)

LabVIEW Type de données	HiQ Type de données	MATLAB Type de données
	Complexe	Complexe
	Vecteur complexe	Vecteur complexe
	Matrice complexe	Matrice complexe

Suggestions de programmation pour les scripts HiQ et MATLAB

Les techniques de programmation suivantes facilitent la mise au point d'un script :

- Écrivez le script et exécutez-le dans HiQ ou dans MATLAB pour le tester et le mettre au point avant de l'importer dans LabVIEW.
- Cliquez sur le nœud de script HiQ avec le bouton droit de la souris et sélectionnez **Éditer dans le serveur** dans le menu local pour éditer, mettre au point, compiler et exécuter dans le script une fenêtre de script HiQ natif.
- Vérifiez les types de données. Lorsque vous créez une nouvelle entrée ou une nouvelle sortie, assurez-vous que le type de données du terminal est correct. À la fois dans HiQ et dans MATLAB, une variable peut changer de type pendant le calcul. Vous pouvez utiliser les fonctions Entrée d'erreur et Sortie d'erreur pour garder une trace de cette opération.
- Créez des commandes et des indicateurs pour les entrées et les sorties afin de pouvoir surveiller les valeurs que les nœuds de script passent de LabVIEW à HiQ ou MATLAB et vice versa. Ceci vous permet de localiser, si besoin est, l'endroit où un nœud de script a calculé une valeur de manière incorrecte.
- Utilisez les paramètres de vérification d'erreur pour la mise au point des informations. Créez un indicateur pour le terminal de **sortie d'erreur** d'un nœud de script pour pouvoir visualiser l'information sur les erreurs pendant l'exécution. Les Boîtes de calcul visualisent les erreurs à la compilation.

Fichiers support HiQ requis avec une application LabVIEW

Après avoir construit une application LabVIEW incluant des appels aux VIs HiQ, vous devez examiner l'ordinateur cible qui exécute l'application.

L'appel au nœud de script HiQ en code LabVIEW lance HiQ. Si vous distribuez une application LabVIEW contenant un script HiQ, HiQ doit être installé sur la machine cible.

La table suivante dresse la liste des fichiers dont vous avez besoin en fonction de la manière dont vous utilisez et distribuez LabVIEW et HiQ.

Fichiers à distribuer	Logiciels actuellement disponibles sur la cible	Que faut-il installer ?
VIs LabVIEW Application avec le nœud de script HiQ	LabVIEW (toutes versions) installé ; pas de HiQ installé	Copie autorisée de HiQ disponible avec LabVIEW
Exécutable	Ni LabVIEW ni HiQ installés	HiQ Reader permet aux autres utilisateurs de visualiser les notebooks professionnels HiQ et d'exécuter des scripts. Vous pouvez distribuer le HiQ Reader librement à partir du CD Système de développement complet ou professionnel de LabVIEW. Vous pouvez télécharger le HiQ Reader à partir du site Web ni.com de National Instruments.

Organisation de LabVIEW

Cette annexe décrit la structure du système de fichiers de LabVIEW et les emplacements conseillés pour l'enregistrement des fichiers.

Organisation de la structure des répertoires de LabVIEW

Cette section décrit la structure du système de fichiers de LabVIEW sous Windows, Macintosh, et UNIX. LabVIEW installe les drivers des matériels GPIB, DAQ, VISA, IVI, Motion Control et IMAQ en fonction de la disponibilité des plates-formes. Reportez-vous au chapitre 3, *Installation et configuration de votre matériel de mesure*, du *LabVIEW Measurements Manual* pour de plus amples informations sur la configuration de votre matériel.

Comme l'expliquent les *Notes d'informations LabVIEW* qui vous sont remises avec vos logiciels, votre répertoire LabVIEW contient les ensembles suivants une fois l'installation terminée.

Bibliothèques

- `user.lib` : contient les commandes et les VIs que vous créez. LabVIEW affiche les commandes dans la palette **Commandes» Commandes utilisateur** et les VIs dans la palette **Fonctions» Bibliothèques utilisateur**. Ce répertoire ne change pas si vous mettez à jour ou supprimez LabVIEW.
- `vi.lib` : contient les bibliothèques des VIs intégrés, dont les VIs GPIB, les VIs d'analyse et les VIs DAQ. LabVIEW affiche ces VIs dans des groupes en relation les uns avec les autres dans la palette **Fonctions**. N'enregistrez pas les fichiers dans `vi.lib` parce que LabVIEW écrase les fichiers dans ce répertoire lorsque vous effectuez une mise à jour.

- `instr.lib` : contient les drivers d'instruments utilisés pour contrôler les instruments PXI, VXI, GPIB, série et les cartes instruments. Lorsque vous installez les drivers d'instruments de National Instruments, placez-les dans ce répertoire. LabVIEW les ajoute à la palette **E/S d'instruments»Drivers d'instruments**.

Structure et support

- `menus` : contient des fichiers que LabVIEW utilise pour configurer la structure des palettes **Commandes** et **Fonctions**.
- `resource` : contient des fichiers support supplémentaires pour l'application LabVIEW. N'enregistrez pas de fichiers dans ce répertoire parce que LabVIEW écrase les fichiers qui s'y trouvent lorsque vous effectuez une mise à jour.
- `project` : contient des fichiers qui deviennent des éléments du menu **Outils** de LabVIEW.
- `templates` : contient des modèles pour des VIs courants.
- `www` : emplacement des fichiers HTML auxquels vous pouvez accéder par l'intermédiaire du serveur Web.

Exercices et exemples

- `activity` : contient les VIs que vous utilisez pour les activités du *Tutorial LabVIEW*. Le répertoire `activity\solution` contient les VIs terminés pour chaque activité.
- `examples` : contient des exemples de VIs. Sélectionnez **Aide»Recherche d'exemples** pour rechercher et parcourir des exemples.

Documentation

- `manuals` : contient la documentation au format PDF. Ce dossier ne contient pas les fichiers d'aide. Accédez à la Bibliothèque LabVIEW en sélectionnant **Aide»Rechercher dans les manuels PDF...**
- `help` : contient les fichiers d'aide. Accédez à l'*Aide LabVIEW* en sélectionnant **Aide»Aide LabVIEW...**

Fichier divers

- `serpdrv` : fichier support pour l'accès au port série sous Windows et UNIX. Distribuez ce fichier avec toutes les applications qui utilisent le port série sur ces plates-formes.

Macintosh

En plus des répertoires mentionnés précédemment, les utilisateurs de Macintosh disposent d'un dossier de bibliothèques partagées qui contient les fichiers support pour l'application LabVIEW.

Emplacement suggéré pour enregistrer les fichiers

LabVIEW installe les répertoires `vi.lib` et `resource` pour les besoins du système LabVIEW uniquement. N'enregistrez pas vos fichiers dans ces répertoires.

Vous pouvez sauvegarder vos fichiers dans les répertoires suivants :

- `user.lib` : Tous les VIs d'utilisation courante que vous voulez voir apparaître dans la palette **Fonctions»Bibliothèques utilisateur**. Utilisez ce répertoire pour étendre les fonctions de `vi.lib`.



Remarque N'enregistrez dans le répertoire `user.lib` que des sous-VIs portables, sans modification, d'un projet à l'autre. Les chemins de VIs dans `user.lib` sont absolus. Les chemins de sous-VIs que vous enregistrez ailleurs sont relatifs au VI parent. Par conséquent, copier un VI de `user.lib` afin de le modifier pour une condition particulière ne change pas le chemin vers ses sous-VIs situés dans `user.lib`.

- `instr.lib` : tout VI de driver d'instruments que vous voulez voir apparaître dans la palette **E/S d'instruments»Drivers d'instruments**.
- `project` : les VIs que vous utilisez pour étendre les fonctions LabVIEW. Les VIs que vous stockez dans ce répertoire s'affichent dans le menu **Outils**.
- `www` : emplacement des fichiers HTML auxquels vous pouvez accéder par le serveur Web.
- `help` : tous les VIs, les fichiers PDF et les fichiers `.hlp` que vous voulez rendre accessibles dans le menu d'**Aide**.

Vous pouvez aussi créer un répertoire n'importe où sur votre disque dur pour stocker les fichiers LabVIEW que vous créez. Reportez-vous à la section [Enregistrement des VIs](#) du chapitre 7, *Création de VIs et de sous-VIs*, pour obtenir de plus amples informations sur les clusters.

Fonctions polymorphes

Les fonctions sont polymorphes à des degrés variables : aucune, certaines ou toutes leurs sorties peuvent être polymorphes. Certaines entrées de fonctions acceptent des valeurs numériques ou booléennes. Certaines acceptent des valeurs numériques ou des chaînes. Certaines acceptent non seulement des valeurs numériques scalaires mais aussi des tableaux de valeurs numériques, des clusters de valeurs numériques et ainsi de suite. Certaines acceptent uniquement des tableaux à une seule dimension, même si les éléments de tableau peuvent être de n'importe quel type. Certaines fonctions acceptent tous les types de données y compris les nombres complexes. Reportez-vous à la note d'application *Polymorphic Units in LabVIEW* pour obtenir de plus amples informations sur les fonctions polymorphes.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur les fonctions polymorphes.

Conversion numérique

Vous pouvez convertir toute représentation numérique en toute autre représentation numérique. Lorsque vous câblez au moins deux entrées numériques de représentations différentes à une fonction, la fonction renvoie habituellement la sortie au format le plus grand ou le plus large. Les fonctions contraignent les petites représentations dans la représentation la plus large avant l'exécution et LabVIEW place un nœud de coercition sur le terminal où la conversion a lieu.

Certaines fonctions, telles que les fonctions Diviser, Sinus et Cosinus, produisent toujours des nombres réels. Si vous câblez des entiers à leurs entrées, ces fonctions vont les convertir en flottants double précision avant d'effectuer le calcul.

Pour représenter une quantité scalaire de type flottant, il est recommandé d'utiliser des flottants double précision. Les nombres à virgule flottante simple précision ne font gagner que peu ou aucun de temps d'exécution et

ils sont plus sujets au débordement. Les bibliothèques d'analyse, par exemple, utilisent des flottants double précision. N'utilisez les flottants à précision étendue que lorsque cela est nécessaire. Les performances et la précision de l'arithmétique de type précision étendue varient suivant les plates-formes. Reportez-vous à la section *Données indéfinies ou inattendues* du chapitre 6, *Exécution et mise au point des VIs*, pour obtenir de plus amples informations sur le débordement des flottants.

Pour représenter des entiers, il est habituellement préférable d'utiliser un entier 32 bits signé.

Si vous câblez une sortie à une destination dont la représentation numérique est différente, LabVIEW convertit les données en fonction des règles suivantes :

- **Entier signé ou entier non signé en flottant** : la conversion est exacte, sauf en ce qui concerne la conversion des entiers longs en flottants simple précision. Dans ce cas, LabVIEW réduit la précision de 32 bits à 24 bits.
- **Flottant en entier signé ou en entier non signé** : LabVIEW fait passer les valeurs hors gamme à la valeur minimale ou à la valeur maximale de l'entier. La plupart des objets entier, comme le terminal d'itération d'une boucle For, arrondissent les flottants. LabVIEW arrondit un nombre flottant dont la partie fractionnaire est 0,5 à l'entier pair le plus proche. Ainsi, LabVIEW arrondit 6,5 à 6 plutôt qu'à 7.
- **Entier à entier** : LabVIEW ne fait pas passer les valeurs hors gamme à la valeur minimale ou à la valeur maximale de l'entier. Si la source est plus petite que la destination, LabVIEW étend le signe d'une source signée et place des zéros dans les bits supplémentaires d'une source non signée. Si la source est plus grande que la destination, LabVIEW ne copie que les bits les moins importants de la valeur.

Polymorphisme pour les fonctions numériques

Les fonctions arithmétiques utilisent des données d'entrée numériques. À quelques exceptions près, mentionnées dans les descriptions des fonctions, la sortie a la même représentation numérique que l'entrée, ou que l'entrée la plus large si les entrées ont des représentations différentes.

Les fonctions arithmétiques peuvent être utilisées avec des nombres, des tableaux de nombres, des clusters de nombres, des tableaux de clusters de nombres, des nombres complexes, et ainsi de suite. Le type de données autorisé peut être défini de manière formelle et récursive comme suit :

Type numérique = scalaire numérique OU tableau [*type numérique*]
OU cluster [*types numériques*]

Les scalaires numériques peuvent être des nombres à virgule flottante, des entiers ou des nombres complexes à virgule flottante. LabVIEW ne vous permet pas d'utiliser des tableaux de tableaux.

Les tableaux peuvent avoir n'importe quel nombre de dimensions, lesquelles peuvent être de toute taille. Les clusters peuvent avoir n'importe quel nombre d'éléments. Le type de sortie des fonctions a la même représentation numérique que le type d'entrée. En ce qui concerne les fonctions à une seule entrée, les fonctions traitent chaque élément du tableau ou du cluster.

Avec les fonctions à deux entrées, vous pouvez utiliser les combinaisons d'entrées suivantes :

- **Similaire** : les deux entrées ont la même structure et la sortie a la même structure que les entrées.
- **Un scalaire** : une des entrées est un scalaire numérique, l'autre est un tableau ou un cluster et la sortie est un tableau ou un cluster.
- **Tableau de** : une des entrées est un tableau numérique, l'autre est le type numérique lui-même et la sortie est un tableau.

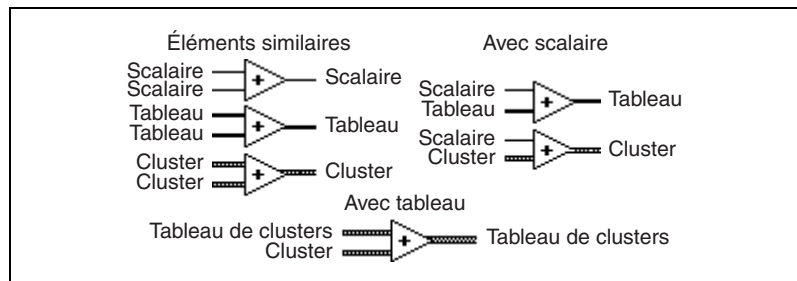
Dans le cas d'entrées similaires, LabVIEW exécute la fonction sur chaque élément des structures. Ainsi, LabVIEW peut ajouter deux tableaux élément par élément. Les deux tableaux doivent être de la même dimension. Si vous ajoutez des tableaux dont les nombres d'éléments diffèrent, la sortie de l'addition aura le même nombre d'éléments que la plus petite entrée. Les clusters doivent avoir le même nombre d'éléments, et les éléments doivent tous être du même type.

Vous ne pouvez pas utiliser la fonction Multiplier pour effectuer une multiplication de matrice. Si vous utilisez la fonction Multiplier avec deux matrices, LabVIEW prend le premier nombre de la première ligne de la première matrice, le multiplie par le premier nombre de la première ligne de la deuxième matrice, et ainsi de suite.

Pour les opérations impliquant un scalaire et un tableau ou un cluster, LabVIEW exécute la fonction sur le scalaire et sur les éléments respectifs de la structure. Ainsi, LabVIEW peut soustraire un nombre de tous les éléments d'un tableau, sans tenir compte de la dimension du tableau.

Pour les opérations impliquant un type numérique et un tableau de ce type, LabVIEW exécute la fonction sur chaque élément du tableau. Ainsi, un graphe est un tableau de points et un point est un cluster de deux types numériques, x et y . Pour décaler un graphe de cinq unités dans la direction de x et de huit unités dans la direction de y , vous pouvez ajouter le point (5, 8), au graphe.

L'exemple ci-dessous illustre les possibilités de combinaisons polymorphes de la fonction Ajouter.



Polymorphisme pour les fonctions booléennes

Les fonctions logiques utilisent des données d'entrée booléennes ou des données d'entrée numériques. Si l'entrée est numérique, LabVIEW effectue l'opération en fonction des bits. Si l'entrée est un entier, la sortie a la même représentation. Si l'entrée est un nombre à virgule flottante, LabVIEW l'arrondit à un entier long et le type de la sortie est un entier long.

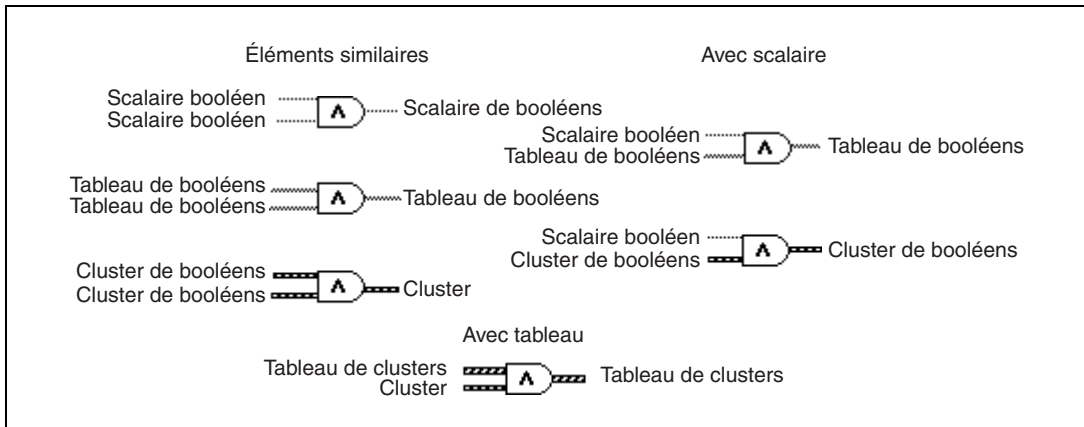
Les fonctions logiques traitent les tableaux de nombres ou de valeurs booléennes, les clusters de nombres ou de valeurs booléennes, les tableaux de clusters de nombres ou de valeurs booléennes, et ainsi de suite.

Le type de données autorisé peut être défini de manière formelle et récursive comme suit :

Type logique = scalaire booléen OU scalaire numérique OU tableau
[*type logique*] OU cluster [*types logiques*]

exceptés les nombres complexes et les tableaux de tableaux qui ne sont pas autorisés.

Les fonctions logiques à deux entrées peuvent avoir les mêmes combinaisons d'entrée que les fonctions arithmétiques. Cependant, les opérations de base des fonctions logiques ne peuvent s'effectuer qu'entre deux valeurs booléennes ou deux nombres. Ainsi, vous ne pouvez pas avoir un ET entre une valeur booléenne et un nombre. L'exemple ci-dessous présente des combinaisons de valeurs booléennes pour la fonction ET.



Polymorphisme pour les fonctions de tableaux

La plupart des fonctions de tableau acceptent les tableaux à n dimensions de tout type. Cependant, les diagrammes de câblage des descriptions des fonctions affichent les tableaux numériques comme type de données par défaut.

Polymorphisme pour les fonctions de chaîne

Les fonctions Longueur d'une chaîne, En majuscules, En minuscules, Renverser une chaîne et Rotation dans une chaîne acceptent les chaînes, les clusters et les tableaux de chaînes, et les tableaux de clusters. Les fonctions En majuscules et En minuscules acceptent aussi les nombres, les clusters de nombres et les tableaux de nombres, en les interprétant comme des caractères de code ASCII. Les entrées de largeur et de précision doivent être scalaires.

Polymorphisme pour les fonctions de conversion de chaînes

Les fonctions Chemin en chaîne et Chaîne en chemin sont polymorphes. Par conséquent, elles traitent les valeurs scalaires, les tableaux de scalaires, les clusters de scalaires, les tableaux de clusters de scalaires, et ainsi de suite. La sortie a la même composition que l'entrée, mais elle est du nouveau type.

Polymorphisme pour les fonctions Chaîne en nombre supplémentaires

Les fonctions Nombre en chaîne décimale, nombre en chaîne hexadécimale, Nombre en chaîne octale, Nombre en chaîne ingénieur, Nombre en chaîne fractionnaire et Nombre en chaîne exponentielle acceptent les clusters et les tableaux de nombres et produisent des clusters et des tableaux de chaînes. Chaîne décimale en nombre, Chaîne hexadécimale en nombre, Chaîne octale en nombre et Chaîne exponentiel/Fract/Sci en nombre acceptent les clusters et les tableaux de chaînes et produisent des clusters et des tableaux de nombres. Les entrées de largeur et de précision doivent être scalaires.

Polymorphisme pour les fonctions de cluster

Les fonctions Assembler et Désassembler n'affichent pas le type de données de leurs terminaux d'entrée et de sortie tant que vous ne câblez pas d'objets à ces terminaux. Lorsque vous les câblez, ces terminaux ont une apparence similaire à celle des types de données des terminaux de la commande ou de l'indicateur de face-avant correspondants.

Polymorphisme pour les fonctions de comparaison

Les fonctions Égale ?, Non Égale ? et Sélectionner utilisent des entrées de tout type, mais les entrées de chaque fonction doivent être du même type.

Les fonctions Supérieur ou Égal ?, Inférieur ou Égal ?, Inférieur? , Supérieur ?, Max & Min et Dans la gamme ? acceptent des entrées de tout type sauf les complexes, les chemins et les refnums, mais les entrées de chaque fonction doivent être du même type. Vous pouvez comparer des nombres, des chaînes, des booléens, des tableaux de chaînes, des clusters de nombres, des clusters de chaînes, et ainsi de suite. Cependant, vous ne pouvez pas comparer un nombre à une chaîne ou une chaîne à un booléen, et ainsi de suite.

Les fonctions qui comparent des valeurs à zéro acceptent les scalaires numériques, les clusters et les tableaux de nombres. Ces fonctions sortent des valeurs booléennes de même structure de données que l'entrée.

La fonction Pas un nombre/chemin/refnum accepte les mêmes types d'entrée que les fonctions qui comparent des valeurs à zéro. Cette fonction accepte aussi les chemins et les refnums. La fonction Pas un nombre/chemin/refnum sort des valeurs booléennes de même structure de données que l'entrée.

Les fonctions Chiffre décimal ?, Chiffre hexa ?, Chiffre octal ?, Imprimable ? et Espace blanc ? acceptent une chaîne scalaire ou un nombre en entrée, des clusters de chaînes ou des nombres non complexes, des tableaux de chaînes ou des nombres non complexes, et ainsi de suite. La sortie se compose de valeurs booléennes de même structure de données que l'entrée.

La fonction Chemin/Chaîne vide ? accepte un chemin, une chaîne scalaire, des clusters de chaînes, des tableaux de chaînes, et ainsi de suite. La sortie se compose de valeurs booléennes de même structure de données que l'entrée.

Vous pouvez utiliser les fonctions Égax ?, Non Égax ?, Pas un nombre/chemin/refnum ?, Chemin/Chaîne vide ? et Sélectionner avec des chemins et des refnums, mais aucune autre fonction de comparaison n'accepte les chemins et les refnums comme entrées.

Les fonctions de comparaison qui utilisent des tableaux et des clusters produisent habituellement des tableaux et des clusters de booléens de même structure. Vous pouvez cliquer la fonction avec le bouton droit de la souris et sélectionner **Comparer des ensembles**, auquel cas la fonction sort une seule valeur booléenne. Cette fonction compare le premier ensemble d'éléments pour produire la sortie, à moins que les premiers éléments ne soient égaux, auquel cas la fonction compare le deuxième ensemble d'éléments, et ainsi de suite.

Polymorphisme pour les fonctions journal

Les fonctions logarithmiques acceptent des données numériques en entrée. Si l'entrée est un entier, la sortie est un flottant double précision. Sinon, la sortie a la même représentation numérique que l'entrée.

Ces fonctions traitent les nombres, les tableaux de nombres, les clusters de nombres, les tableaux de clusters de nombres, les nombres complexes, et ainsi de suite. Le type de données autorisé peut être défini de manière formelle et récursive comme suit :

Type numérique = scalaire numérique OU tableau [type numérique]
OU cluster [types numériques]

mais les tableaux de tableaux ne sont pas autorisés.

Les tableaux peuvent être de toute taille et peuvent comporter n'importe quel nombre de dimensions. Les clusters peuvent avoir n'importe quel nombre d'éléments. Le type de sortie est de la même représentation numérique que l'entrée, et les fonctions s'exécutent sur chaque élément du cluster ou du tableau. Reportez-vous à la section [Polymorphisme pour les fonctions numériques](#) de ce chapitre pour obtenir de plus amples informations sur les fonctions polymorphes à deux entrées. Les combinaisons autorisées de types d'entrée pour les fonctions logarithmiques à deux entrées peuvent être les suivantes :

- **Similaire** : les deux entrées ont la même structure et la sortie a la même structure que les entrées.
- **Un scalaire** : l'une des entrées est un scalaire numérique, l'autre est un tableau ou un cluster numérique et la sortie est un tableau ou un cluster.



Fonctions de comparaison

Utilisez les fonctions de comparaison qui se trouvent sur la palette **Fonctions»Comparaison** afin de comparer des valeurs booléennes, des chaînes de caractères, des valeurs numériques, des tableaux et des clusters. La plupart des fonctions de comparaison testent une entrée ou comparent deux entrées et renvoient une valeur booléenne.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur les fonctions de comparaison.

Comparaison de valeurs booléennes

Les fonctions de comparaison traitent la valeur booléenne VRAI en la considérant comme supérieure à la valeur booléenne FAUX.

Comparaison de chaînes

LabVIEW compare des chaînes de caractères en se basant sur l'équivalent numérique des caractères de code ASCII. Par exemple, a (de valeur décimale 97) est supérieur à A (65), qui est supérieur au chiffre 0 (48), qui est supérieur au caractère espace (32). LabVIEW compare les caractères un par un à partir du début de la chaîne jusqu'à ce qu'il trouve une différence, auquel cas il interrompt la comparaison. Par exemple, LabVIEW évalue les chaînes abcd et abef jusqu'à ce qu'il trouve c, qui est inférieur à la valeur de e. Lorsqu'une chaîne comporte un caractère de plus, elle est considérée comme supérieure à la deuxième chaîne. Ainsi, la chaîne abcd est supérieure à abc parce que la première chaîne est plus longue.

Les fonctions qui testent la catégorie d'un caractère de chaîne, comme les fonctions Chiffre décimal ? et Imprimable ? n'évaluent que le premier caractère de la chaîne.

Comparaison de valeurs numériques

Les fonctions de comparaison convertissent les valeurs numériques dans la même représentation avant de les comparer. Les comparaisons avec une ou deux entrées comportant la valeur Pas un nombre (NaN) renvoient une valeur indiquant une différence. Reportez-vous à la section *Données indéfinies ou inattendues* du chapitre 6, *Exécution et mise au point des VIs*, pour obtenir de plus amples informations sur la valeur NaN.

Comparaison des tableaux et clusters

Certaines fonctions de comparaison comportent deux modes de comparaison de tableaux et de clusters de données. En mode Comparer des ensembles, la fonction renvoie une valeur scalaire simple lorsque vous comparez deux tableaux ou deux clusters. En mode Comparer des éléments, la fonction compare les éléments un à un et renvoie un tableau ou un cluster de valeurs booléennes.

En mode Comparer des ensembles, les opérations de comparaison de chaînes et de comparaison de tableaux suivent exactement la même procédure, à savoir la chaîne est traitée comme un tableau de caractères de code ASCII.

Certaines fonctions de comparaison ne fonctionnent qu'en mode Comparer des ensembles et, par conséquent, les options de menu local ne s'affichent pas.

Tableaux

Lorsque vous comparez des tableaux multidimensionnels, tous les tableaux entrés dans la fonction doivent être de la même dimension. Les fonctions de comparaison qui ne sont pas munies des modes Comparer des ensembles et Comparer des éléments, comparent les tableaux de la même manière qu'elles comparent les chaînes, c'est-à-dire un élément à la fois jusqu'à ce qu'elles rencontrent une inégalité et en commençant par le premier élément.

Mode Comparer des éléments

En mode Comparer des éléments, les fonctions de comparaison sortent un tableau de valeurs booléennes aux mêmes dimensions que les tableaux d'entrée. Chaque dimension du tableau de sortie est de la taille du plus petit des deux tableaux d'entrée dans cette dimension. Au même titre que pour chaque dimension (ligne, colonne, ou page), les fonctions comparent les

valeurs des éléments correspondants dans chaque tableau d'entrée pour produire la valeur booléenne correspondante dans le tableau de sortie.

Mode Comparer des ensembles

En mode Comparer des ensembles, les fonctions de comparaison renvoient comme résultat un booléen unique. LabVIEW considère les valeurs correspondantes récentes qui s'affichent dans les tableaux d'entrée comme secondaires aux valeurs anciennes. LabVIEW effectue les opérations suivantes pour déterminer le résultat de la comparaison :

- LabVIEW compare les éléments correspondants dans chaque tableau d'entrée, en partant du début du tableau.
- Si les éléments correspondants *ne sont pas égaux*, LabVIEW s'interrompt et le résultat de cette comparaison est utilisé comme sortie de la fonction de comparaison.
- Si les éléments sont égaux, LabVIEW traite la paire de valeurs suivante, jusqu'à ce qu'il trouve une inégalité ou atteigne la fin de l'un des tableaux d'entrée.
- Si toutes les valeurs dans les tableaux d'entrée sont égales mais que l'un d'entre eux comporte des éléments supplémentaires, le tableau le plus long est considéré comme étant supérieur à l'autre. Par exemple, LabVIEW considère que le tableau [1, 2, 3, 2] est supérieur au tableau [1, 2, 3].

Clusters

Les clusters que vous comparez doivent comporter le même nombre d'éléments, les types d'éléments doivent être compatibles et les éléments doivent être dans le même ordre de cluster. Ainsi, un cluster composé d'une DBL et d'une chaîne peut être comparé à un cluster composé d'un I32 et d'une chaîne.

Mode Comparer des éléments

En mode Comparer des éléments, les fonctions de comparaison renvoient un cluster d'éléments booléens, un pour chaque élément correspondant dans les clusters d'entrée.

Mode Comparer des ensembles

En mode Comparer des ensembles, les fonctions de comparaison renvoient comme résultat un booléen unique. LabVIEW compare des éléments correspondants jusqu'à ce qu'il trouve une inégalité qui va déterminer le résultat. La fonction ne va considérer que les deux clusters sont égaux que si tous les éléments sont égaux.

Utilisez le mode Comparer des ensembles sur des clusters si vous comparez deux enregistrements contenant des données classées, dans lesquels les éléments récents du cluster sont considérés comme des clés secondaires par rapport aux éléments plus anciens. Par exemple, un cluster contenant deux chaînes, une pour le nom de famille suivi d'une autre pour le prénom, ne comparerait les champs correspondant au prénom que si les champs correspondant au nom de famille concordent.

Masquage des données numériques

Dans le diagramme, créez un masque en assemblant la waveform numérique avec un tableau 2D d'entiers, comme illustré dans la figure D-1.

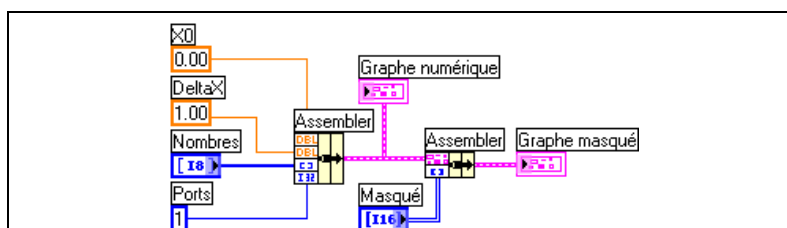


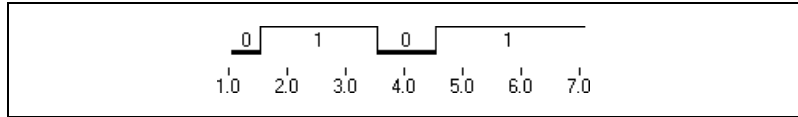
Figure D-1. Masquage des données numériques

Combinez des bits en spécifiant des valeurs dans chaque ligne du tableau 2D **Masque**.

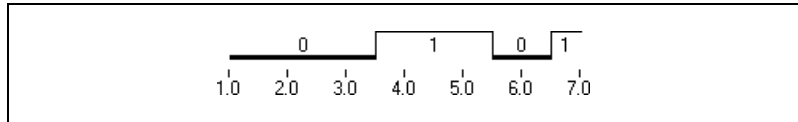
Par exemple, considérons un tableau 1D composé de sept nombres 8 bits : 1, 2, 7, 32, 55, 82 et 127. La table ci-dessous affiche les représentations binaires de ces nombres.

	1	2	7	32	55	82	127
Bit 0	1	0	1	0	1	0	1
Bit 1	0	1	1	0	1	1	1
Bit 2	0	0	1	0	1	0	1
Bit 3	0	0	0	0	0	0	1
Bit 4	0	0	0	0	1	1	1
Bit 5	0	0	0	1	1	0	1
Bit 6	0	0	0	0	0	1	1
Bit 7	0	0	0	0	0	0	0

En utilisant un masque, vous pouvez tracer un bit sur un graphe et superposer un autre bit sur le même tracé. Par exemple, le bit 1 comporte les valeurs 0, 1, 1, 0, 1, 1 et 1, comme illustré dans la courbe suivante.



Le bit 5 comporte les valeurs 0, 0, 0, 1, 1, 0 et 1, comme illustré dans la courbe suivante.



Un masque correspond à un tableau 2D. Chaque ligne du tableau représente une courbe dans le graphe numérique. Pour afficher le bit 1 et le bit 5 sur la courbe, entrez 1 dans l'un des éléments du tableau et 5 dans un autre élément, comme illustré dans la figure D-2. Entrez -1 dans les éléments du tableau qui ne doivent pas représenter de bit.

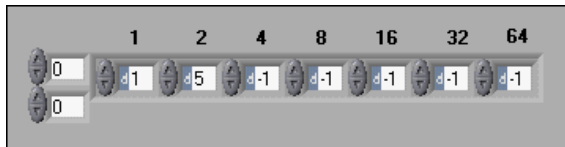
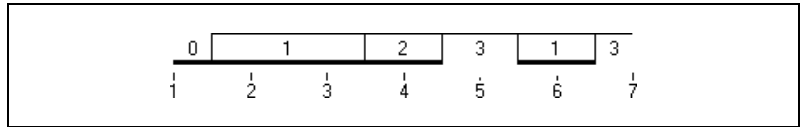


Figure D-2. Exemple d'une commande de masquage de tableau

Dans cet exemple, chaque élément du tableau est identifié par une puissance de deux (2^0 à 2^6). Utilisez ces nombres pour identifier quel élément du tableau représente un bit dans le graphe. Dans cet exemple, l'élément 1 représente le bit 1 dans le graphe, de telle sorte que toute valeur de 1 dans le bit assigne un 1 à la courbe, et toute valeur de 0 assigne un 0 à la courbe. L'élément 2 représente le bit 5 dans le graphe, de telle sorte que toute valeur de 1 dans le bit assigne un 2 à la courbe et toute valeur de 0 assigne un 0 à la courbe.

Le tableau de la figure D-2 donne comme résultat la courbe ci-après dans le graphe numérique.



Cette courbe fournit les indications suivantes :

- Le bit 1 et le bit 5 du premier élément comprennent une valeur de 0.
- Le bit 1 comprend une valeur de 1 dans les deuxième et troisième éléments. Le bit 5 comprend une valeur de 0 dans les deuxième et troisième éléments.
- Le bit 1 comprend une valeur de 0 dans le quatrième élément. Le bit 5 comprend une valeur de 1 dans le quatrième élément.
- Le bit 1 et le bit 5 du cinquième élément comprennent une valeur de 1.
- Le bit 1 du sixième élément comprend une valeur de 1. Le bit 5 du sixième élément comprend une valeur de 0.
- Le bit 1 et le bit 5 du septième élément comprennent une valeur de 1.

Les nombres qui s'affichent dans la courbe découlent de l'équation suivante :

$$\frac{\begin{aligned} &[\text{valeur du bit (1 ou 0)}] \times 2^{[\text{numéro de l'élément}]} \\ &+ [\text{autre valeur de bit (1 ou 0)}] \times 2^{[\text{autre numéro d'élément}]} \\ &\quad \vdots \\ &+ [\text{valeur du bit } n \text{ (1 ou 0)}] \times 2^{[\text{numéro de l'élément } n]} \end{aligned}}{\text{valeur du point sur la courbe}}$$

Dans la figure D-3, le quatrième (8) et le sixième (32) éléments représentent respectivement le bit 5 et le bit 1.

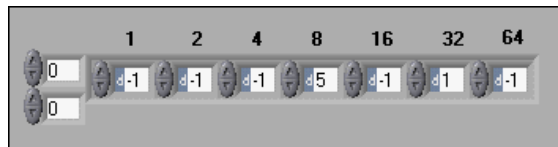
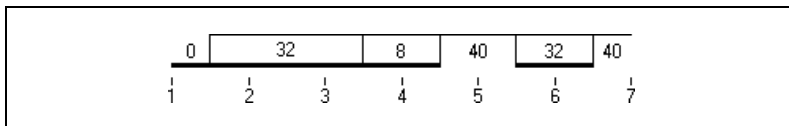


Figure D-3. Spécification des bits identiques dans des éléments différents

Le tableau de la figure D-3 aboutit à la courbe suivante. Il est à noter que les courbes sont similaires mais que les valeurs numériques sont différentes.



Chaque rang du masque correspond à une courbe du graphe. Ajoutez un nouveau rang au masque pour ajouter une nouvelle courbe au graphe numérique, comme illustré dans la figure D-4.

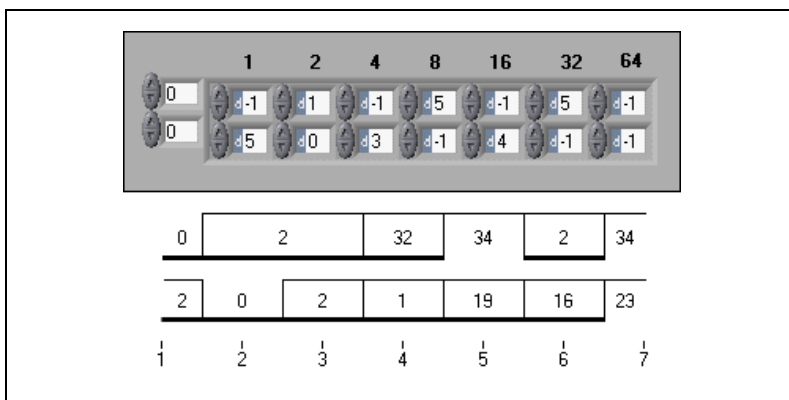


Figure D-4. Création de deux courbes de données masquées sur un graphe numérique

Ressources techniques

Support sur le Web

Le site Web de National Instruments est la première source à consulter pour répondre à vos questions et résoudre vos problèmes en matière d'installation, de configuration et de développement d'applications. Ces ressources en ligne pour le diagnostic et la résolution de problèmes comprennent des questions couramment posées (FAQ), des bases de connaissances (Knowledge Bases), des assistants de dépannage spécialisés par ligne de produit, des manuels, des drivers, des mises à jour de logiciels et bien d'autres choses encore. Ce support Web est disponible à la rubrique Technical Support du site ni.com.

NI Developer Zone

NI Developer Zone, accessible à l'adresse ni.com/zone, est la ressource principale pour le développement de systèmes de mesure et d'automatisation. Vous pourrez facilement y trouver les derniers exemples de programmes, configureurs de systèmes et tutoriaux, des nouveautés techniques, et entrer en contact avec une communauté de développeurs pour échanger vos connaissances.

Formation

National Instruments propose de nombreuses solutions pour satisfaire vos besoins en formation : depuis des tutoriaux adaptés à votre propre rythme, des vidéos, des CD interactifs, jusqu'aux cours dispensés par des instructeurs partout dans le monde. Consultez la rubrique Customer Education du site ni.com pour obtenir des informations sur le calendrier et le programme des cours, les centres de formation et les inscriptions.

Intégration de systèmes

Si vous devez concilier délais serrés, ressources techniques limitées et toute autre contrainte, nous vous invitons à faire appel aux services d'intégrateurs de systèmes ou de consultants. Vous pourrez alors vous reposer sur le savoir-faire des membres internationaux du Programme Alliance. Pour en savoir plus sur les solutions d'intégration de systèmes proposées par le réseau Alliance, consultez la rubrique System Integration du site ni.com.

Support international

National Instruments possède des filiales dans le monde entier pour satisfaire vos besoins en matière de support technique. Vous pouvez accéder aux sites Web de nos filiales à partir de la rubrique Worldwide Offices du site ni.com. Ces sites vous fournissent les dernières informations pour contacter les filiales, les coordonnées du support technique local, les adresses e-mail et la liste des événements en cours ou à venir dans chaque pays.

Si vous n'avez pas pu trouver les réponses à vos questions en parcourant les ressources techniques mises à votre disposition sur notre site Web, contactez la filiale National Instruments la plus proche. Les numéros de téléphone de nos filiales figurent au début de ce manuel.

Glossaire

Préfixe	Signification	Valeur
m-	milli-	10^{-3}
k-	kilo-	10^3
M-	méga-	10^6

Nombres/Symboles

Δ	Delta ; différence. Δx correspond à la valeur par laquelle x passe d'un indice au suivant.
π	Pi.
∞	Infini.
1D	À une dimension.
2D	À deux dimension.

A

A	Ampère.
À deux dimensions	Objet ayant deux dimensions, comme c'est le cas pour un tableau muni de plusieurs lignes et de plusieurs colonnes.
À une dimension	Objet ayant une dimension, comme c'est le cas pour un tableau muni d'une ligne d'éléments.
acquisition de données	Procédé d'acquisition des données, généralement à partir de périphériques enfichables d'entrée analogique/numérique ou numérique.
application multithread	Application qui exécute plusieurs threads d'exécution différents de manière indépendante. Sur un ordinateur à plusieurs processeurs, les différents threads peuvent s'exécuter simultanément sur des processeurs différents.
ASCII	American Standard Code for Information Interchange.

Assistant d'E/S DAQ	Utilitaire servant à nommer et à configurer les voies DAQ analogiques et numériques. Disponible dans le voisinage de données Measurement & Automation Explorer (Windows) ou DAQ Channel Wizard (Macintosh).
auto-indexation	Capacité des structures de boucle à désassembler et à assembler des tableaux à leurs bordures. Au fur et à mesure qu'un tableau entre dans une boucle avec l'auto-indexation activée, la boucle le désassemble automatiquement en extrayant les scalaires des tableaux ID, les tableaux ID des tableaux 2D, et ainsi de suite. Les boucles assemblent les données en tableaux au fur et à mesure que les données sortent de la boucle en ordre inverse.

B

barre de menus	Barre horizontale qui liste les noms des principaux menus d'une application. La barre de menus s'affiche sous la barre de titre d'une fenêtre. Chaque application est munie d'une barre de menus qui lui est propre, mais certains menus et certaines commandes sont communs à de nombreuses applications.
barre d'outils	Barre contenant les boutons de commande utilisés pour exécuter les VIs et les mettre au point.
bibliothèque de VIs	Fichier spécial contenant une collection de VIs apparentés destinés à une utilisation particulière.
boîte de calcul	Nœud qui exécute des équations que vous entrez sous forme littérale. Particulièrement utile pour les équations très longues, trop encombrantes pour être construites sous forme de diagramme.
boucle For	Structure de boucle itérative qui exécute son sous-diagramme un nombre de fois défini. Équivalent à un code textuel : <code>For i = 0 to n - 1, do...</code>
boucle While	Structure de boucle qui répète une section de code jusqu'à ce qu'une condition soit remplie.
bouton Exécution interrompue	Bouton qui remplace le bouton Exécuter lorsqu'un VI ne peut pas s'exécuter parce que des erreurs se sont produites.
bouts de fils de liaison	Fils de liaison tronqués qui apparaissent à côté de terminaux non câblés lorsque vous déplacez l'outil Bobine sur un VI ou un nœud de fonction.

branche d'un fil de liaison	Section de fil de liaison qui contient tous les segments de fils de liaison, de jonction en jonction, de terminal en jonction ou de terminal en terminal s'il n'y a pas de jonction entre les fils de liaison.
buffer	Zone de stockage temporaire pour données acquises ou générées.
buffer de mémoire	<i>Voir</i> buffer.

C

CA	Courant alternatif.
cadre	Sous-diagramme d'une structure Séquence.
carte/courbe d'intensité	Méthode d'affichage en trois dimensions de données sur une courbe 2D par utilisation de la couleur.
case à cocher	Petite boîte carrée dans une boîte de dialogue que vous pouvez sélectionner ou réinitialiser. Les cases à cocher sont généralement associées à plusieurs options que vous pouvez définir. Vous pouvez sélectionner plus d'une case à cocher.
chaîne	Représentation d'une valeur sous forme de texte.
chemin absolu	Chemin du fichier ou du répertoire qui décrit l'emplacement relatif au niveau principal du système de fichier.
CIN	<i>Voir</i> Code Interface Node (CIN) .
classe du VI	Référence à un instrument virtuel qui autorise l'accès aux propriétés et aux méthodes du VI.
clonage	Utilisé pour faire la copie d'une commande ou celle d'un autre objet en cliquant dessus tout en appuyant sur la touche <Ctrl> et en faisant glisser la copie à l'endroit où vous voulez la placer. (Macintosh) Appuyez sur la touche <Option>. (Sun) Appuyez sur la touche <Meta>. (Linux) Appuyez sur la touche <Alt>. (UNIX) Vous pouvez aussi copier un objet en cliquant dessus avec le bouton central de la souris et en faisant glisser la copie à l'endroit où vous voulez la placer.

cluster	Ensemble d'éléments de données ordonnées et non indexées de tout type de données, notamment les types numérique, booléen, chaîne, tableau ou cluster. Les éléments doivent tous être des commandes ou des indicateurs.
Code Interface Node (CIN)	Nœud particulier du diagramme par lequel vous pouvez relier un code textuel à un VI.
coercition	Conversion effectuée automatiquement par LabVIEW pour changer la représentation numérique d'un élément de données.
commande	Objet de la face-avant utilisé pour entrer des données de manière interactive dans un VI ou par programmation dans un sous-VI, tel qu'un bouton rotatif, un bouton-poussoir ou un cadran.
commande de graphe	Objet de la face-avant qui affiche des données dans un plan cartésien.
commande de menu déroulant	Commande numérique spéciale qui associe des entiers 32 bits à une série d'étiquettes texte ou de graphiques, en commençant à zéro et en augmentant de manière séquentielle.
commandes et indicateurs de chaîne	Objets de la face-avant utilisés pour manipuler et afficher du texte.
commandes et indicateurs booléens	Objets de la face-avant utilisés pour manipuler et afficher des données booléennes (VRAI ou FAUX).
commandes et indicateurs numériques	Objets de la face-avant utilisés pour manipuler et afficher des données numériques.
compilation	Opération qui convertit un code de haut niveau en code exécutable sur machine. LabVIEW compile automatiquement les VIs avant leur première exécution et après leur création ou l'édition d'une modification.
condition	L'un des sous-diagrammes d'une structure Condition.
connecteur	Partie du VI ou du nœud de fonction qui contient les terminaux d'entrée et de sortie. Les données circulent vers ou à partir d'un nœud via un connecteur. Zone dans le coin supérieur droit d'une face-avant ou d'une fenêtre de diagramme qui affiche le motif du terminal du VI. Il définit les entrées et les sorties que vous pouvez câbler à un VI.
constante	Voir constante universelle et constante définie par l'utilisateur .

constante définie par l'utilisateur	Objet de diagramme qui émet une valeur que vous définissez.
constante universelle	Objet de diagramme non éditable qui correspond à un caractère ASCII particulier ou à une constante numérique standard, π par exemple.
conversion	Changer le type d'un élément de données.
coordonnées absolues	Coordonnées d'image relatives à la position (0,0) de l'indicateur d'images.
coordonnées relatives	Coordonnées d'image relatives à la position courante du crayon.
coude sur un fil de liaison	Point qui correspond à la jonction de deux segments.
courbe	Représentation graphique d'un tableau de données sur un graphe ou un graphe déroulant.
courbe en 3D	Courbe paramétrique spéciale $(x(t), y(t), z(t))$, dans laquelle le paramètre t s'exécute sur un intervalle donné.

D

DAQ	Voir acquisition de données .
DDE	Voir dynamic data exchange .
défaut	Valeur prédéfinie. De nombreuses entrées du VI utilisent une valeur par défaut si vous ne spécifiez pas de valeur.
définition de type	Copie principale d'un objet personnalisé que plusieurs VIs peuvent utiliser.
dépendance de données artificielle	Condition en langage de programmation de flux des données dans laquelle l'arrivée des données plutôt que leur valeur déclenche l'exécution d'un nœud.
dépendance des données	Condition en langage de programmation de flux des données dans laquelle un nœud ne peut pas s'exécuter tant qu'il ne reçoit pas de données en provenance d'un autre nœud. Voir aussi dépendance de données artificielle.

diagramme	Description en images ou représentation d'un programme ou d'un algorithme. Le diagramme se compose d'icônes exécutables, appelées nœuds, et de fils de liaison, qui acheminent les données entre les nœuds. Le diagramme représente le code source du VI. Le diagramme se trouve dans la fenêtre de diagramme du VI.
discret	Contient des valeurs discontinues de la variable indépendante, habituellement le temps.
dithering	Ajout d'un bruit gaussien à un signal d'entrée analogique. Vous pouvez augmenter la résolution d'un demi-bit en ajoutant un dither, puis en effectuant la moyenne des données d'entrée.
DLL	Dynamic Link Library.
données aplaties	Données de tout type qui ont été converties en chaîne, habituellement pour pouvoir être écrites dans un fichier.
données waveform	Représentent des lectures multiples effectuées sur un signal à une fréquence d'échantillonnage particulière.
driver	Logiciel de commande d'un périphérique particulier, tel qu'un périphérique DAQ.
drivers d'instrument	VI qui contrôle un instrument programmable.
dynamic data exchange	DDE : voir échange dynamique de données.

E

E/S	Entrée/Sortie. Le transfert des données vers ou à partir d'un système informatique comportant des voies de communication, des périphériques d'entrée opérateur et/ou des interfaces d'acquisition et de contrôle de données.
échantillon	Point unique de données d'entrée ou de sortie analogique ou numérique.
échelle	Partie de graphe déroulant, de graphe et de commandes et d'indicateurs numériques qui contient une série de repères ou de points situés à des intervalles connus pour représenter les unités de mesure.
entier	Tous les nombres entiers relatifs.

entrée d'erreur	Structure d'erreur qui entre dans un VI.
étiquette	Objet texte utilisé pour nommer ou décrire des objets ou des zones sur la face-avant ou le diagramme.
étiquette libre	Étiquette de la face-avant ou du diagramme qui n'appartient à aucun autre objet.
événement	Condition ou état d'un signal analogique ou numérique.

F

face-avant	Interface utilisateur interactive d'un VI. L'affichage de la face-avant simule des instruments physiques, tels que des oscilloscopes et des multimètres.
faire glisser	Utilisation du curseur sur l'écran pour sélectionner, déplacer, copier ou supprimer des objets.
fenêtre active	Fenêtre qui accepte l'entrée utilisateur, généralement la fenêtre de devant. La barre de titres d'une fenêtre active est en surbrillance. Pour activer une fenêtre, cliquez dessus ou sélectionnez-la dans le menu Fenêtres .
fenêtre Aide contextuelle	Fenêtre spéciale de LabVIEW qui affiche les noms et les positions des terminaux pour un VI ou une fonction, la description des commandes et des indicateurs, les valeurs des constantes universelles, de même que les descriptions et les types de données des attributs de commande.
Fenêtre de hiérarchie	Fenêtre qui affiche graphiquement la hiérarchie des VIs et des sous-VIs.
fenêtre de la face-avant	Fenêtre du VI qui contient la face-avant, la barre d'outils, le cadre d'icônes et le connecteur.
fichier journal	Fichier qui stocke les données comme une séquence d'enregistrements de type de données simples et arbitraires que vous spécifiez lorsque vous créez le fichier. Tous les enregistrements d'un fichier journal doivent être d'un seul type, mais ce type peut être complexe. Par exemple, vous pouvez spécifier chaque enregistrement en tant que cluster contenant une chaîne, un nombre et un tableau.
fichier standard	Fichier qui stocke les données comme une séquence de caractères ou d'octets ASCII.
FIFO	Buffer de mémoire premier entré/premier sorti. Les premières données stockées sont les premières données envoyées au destinataire.

fil de liaison	Chemin des données entre les nœuds.
filtrage	Type de conditionnement du signal qui vous permet de filtrer des signaux indésirables en provenance du signal que vous essayez de mesurer.
flux de commande	Système de programmation dans lequel l'ordre séquentiel des instructions détermine l'ordre des exécutions. La plupart des langages de programmation textuels sont des langages de flux de commande.
flux de données	Système de programmation se composant de nœuds exécutables qui ne s'exécutent que lorsqu'ils reçoivent toutes les données d'entrée requises et qui produisent automatiquement une sortie quand ils s'exécutent. LabVIEW est un système par flux de données.
fonction	Élément d'exécution intégré, comparable à un opérateur, une fonction ou une expression en langage de programmation textuel.

G

gamme	Zone à l'intérieur de laquelle une quantité est mesurée, reçue ou transmise. Exprimée en formulant les valeurs de la gamme inférieure et de la gamme supérieure, qui font référence aux limites de la zone.
General Purpose Interface Bus	GPIB : synonyme de HP-IB. Le bus standard utilisé pour contrôler des instruments électroniques par ordinateur. Aussi appelé bus IEEE 488 parce qu'il est défini par les normes ANSI/IEEE 488-1978, 488.1-1987, et 488.2-1992.
glissière	Partie déplaçable des commandes et indicateurs de glissière.
GPIB	<i>Voir</i> General Purpose Interface Bus.
graphe	Affichage 2D d'au moins une courbe. Un graphe reçoit et trace des données.
graphe déroulant	Indicateur qui trace des points de données à une certaine fréquence. Affichage 2D d'au moins une courbe dans laquelle l'affichage conserve les données précédentes jusqu'à un maximum que vous définissez. Le graphe déroulant reçoit les données et met l'affichage à jour point par point ou tableau par tableau, en conservant, pour l'affichage, un certain nombre d'anciens points dans un buffer. Consultez également mode de rafraîchissement .

groupe	<p>Ensemble de voies ou de ports d'entrée ou de sortie que vous définissez. Les groupes peuvent contenir des voies d'entrée analogique, de sortie analogique, d'entrée numérique, de sortie numérique ou de compteur/horloge. Un groupe ne peut contenir qu'un seul type de voie. Utilisez un numéro d'ID de tâche pour faire référence à un groupe après l'avoir créé. Vous pouvez définir jusqu'à 16 groupes en une seule fois.</p> <p>Pour effacer un groupe, câblez un tableau de voie vide et le numéro de groupe au VI de configuration de groupe. Vous n'avez pas besoin d'effacer un groupe pour changer ses éléments. Si vous reconfigurez un groupe dont la tâche est active, LabVIEW efface la tâche et retourne une mise en garde. LabVIEW ne redémarre pas la tâche une fois que vous avez reconfiguré le groupe.</p>
H	
hexa	Hexadécimal. Système de nombres de base 16
I	
icône	Représentation graphique d'un nœud dans un diagramme.
IEEE	Institute for Electrical and Electronic Engineers.
image	Série d'instructions graphiques qu'un indicateur d'images utilise pour créer une image.
impression par programmation	Impression automatique d'une face-avant de VI à la fin de l'exécution.
indicateur	Objet de face-avant qui affiche une sortie, tel qu'un graphe ou une LED.
indicateur image	Indicateur universel pour l'affichage d'images contenant des lignes, des cercles, du texte et d'autres types d'objets graphiques.
Inf	Valeur d'afficheur numérique pour une représentation en flottants de l'infini.
info-bulle	Petit panneau de texte jaune qui identifie le nom du terminal et facilitent l'identification des terminaux pour le câblage.
instrument virtuel (VI)	Programme de LabVIEW dont l'apparence et la fonction sont conçues sur le modèle d'un instrument virtuel.

IP	Protocole Internet (Internet Protocol).
jonction d'un fil de liaison	Point où trois segments minimum de fils de liaison se rejoignent.
J	
journal	Habituellement utilisé pour acquérir des données et les stocker simultanément dans un fichier disque. Les VIs et les fonctions d'E/S sur fichiers de LabVIEW peuvent enregistrer des données.
L	
LabVIEW	Laboratory Virtual Instrument Engineering Workbench. LabVIEW est un langage de programmation graphique qui utilise des icônes et non des lignes de texte pour créer des programmes.
lecteur	Lettre dans la gamme a-z suivie par le signe deux-points (:) pour indiquer un lecteur de disque logique.
LED	Diode électroluminescente.
légende	Objet dont est muni un graphe déroulant ou un graphe pour afficher les noms et les styles de courbe de leurs courbes.
liste déroulante	Boîte à l'intérieur d'une boîte de dialogue qui liste tous les choix disponibles pour une commande. Par exemple, une liste des noms des fichiers sur un disque.
LLB	Bibliothèque de VIs
logiciel d'application	Application créée au moyen du système de développement LabVIEW et exécutée dans l'environnement du système d'exécution LabVIEW.
M	
matrice	Tableau 2D.
Measurement & Automation Explorer	Environnement standard de configuration et de diagnostic du matériel de National Instruments pour Windows.

menu déroulant	Menu auquel vous accédez à partir d'une barre de menus. Les options de menu déroulant sont habituellement des menus de nature générale.
menu local	Menu auquel vous accédez en cliquant sur l'objet avec le bouton droit de la souris. Les options de menu sont spécifiques à l'objet.
message d'erreur	Indique un incident dans les logiciels ou dans le matériel ou indique une tentative d'entrée de données inacceptables.
méthode	Procédure qui s'exécute quand un objet reçoit un message. Une méthode est toujours associée à une classe.
mise à l'échelle automatique	Capacité des échelles à s'ajuster à la gamme des valeurs tracées. Sur les échelles des graphes, la fonction Mise à l'échelle automatique détermine les valeurs d'échelle minimale et maximale.
mode Animation	Technique de mise au point qui consiste à animer l'exécution d'un VI pour illustrer le flux des données dans le VI.
mode de rafraîchissement	Un graphe déroulant dispose de trois modes de mise à jour. graphe déroulant : mode de mise à jour similaire à celui d'un enregistreur papier de graphe déroulant, qui défile au fur et à mesure qu'il trace des données. oscillographe : mode de mise à jour similaire à celui d'un oscilloscope. graphe à balayage : similaire à un oscillographe, sauf qu'une ligne verticale sépare les anciennes données des nouvelles données.
N	
NaN	Valeur d'afficheur numérique pour une représentation en flottants de <i>pas un nombre</i> . Habituellement, le résultat d'une opération non définie, comme $\log(-1)$.
NI-DAQ	Driver complet compris avec tout matériel DAQ de National Instruments.
nœud	Élément d'exécution de programme. Les nœuds sont analogues à des expressions, des opérateurs, des fonctions et des sous-programmes en langage de programmation textuel. Dans un diagramme, les nœuds comprennent les fonctions, les structures et les sous-VIs.

nœud de données d'événement	Nœud attaché aux côtés gauche et droit d'une structure Événement indiquant les données disponibles pour l'événement à gérer par ce cas que vous avez configuré. Si vous configurez un cas unique pour gérer plusieurs événements, seules les données communes à tous les types d'événements gérés seront disponibles.
nœud de propriété	Définit ou recherche les propriétés d'un VI ou d'une application.
nom de la voie	Nom unique donné à une configuration de voies dans l'Assistant Entrées/Sorties DAQ.

O

objet	Terme générique pour tout élément de la face-avant ou du diagramme, notamment les commandes, les indicateurs, les nœuds, les fils de liaison et les images importées.
OLE	Object Linking and Embedding.
outil	Curseur utilisé pour effectuer des opérations particulières.
outil Bobine	Outil utilisé pour définir des chemins de données entre des terminaux.
outil Doigt	Outil utilisé pour entrer des données dans des commandes et les exploiter.
outil Flèche	Outil utilisé pour déplacer et redimensionner des objets.
outil Main	Outil utilisé pour se déplacer d'une fenêtre à l'autre.
outil Menu local	Outil utilisé pour accéder au menu local d'un objet.
outil Pinceau	Outil utilisé pour définir les couleurs de premier plan et d'arrière-plan.
outil Pipette	Copie les couleurs à coller avec l'outil Pinceau.
outil Point d'arrêt	Outil qui sert à définir un point d'arrêt sur un VI, un nœud ou un fil de liaison.
outil Sonde	Outil utilisé pour créer des sondes sur les fils de liaison.
outil Texte	Outil utilisé pour créer des étiquettes et entrer du texte dans des fenêtres texte.

P

palette	Ensemble d'icônes présentant un choix d'options.
palette d'Outils	Palette contenant des outils que vous pouvez utiliser pour éditer et mettre au point des objets de la face-avant et du diagramme.
palette de Commandes	Palette qui contient les commandes, les indicateurs et les objets décoratifs de la face-avant.
palette de Fonctions	Palette qui contient les VIs, les fonctions, les structures du diagramme et les constantes.
périphérique	<p>Instrument ou contrôleur auquel on peut s'adresser comme à une entité unique et qui contrôle ou surveille les points réels d'E/S. Un périphérique peut être relié à l'ordinateur hôte via un type de réseau de communication, ce qui est souvent le cas, ou peut être de type enfichable.</p> <p>Pour les applications d'acquisition de données (DAQ), un périphérique DAQ est situé à l'intérieur de votre ordinateur ou directement connecté au port parallèle de votre ordinateur. Des cartes enfichables, des cartes PCMCIA, et des périphériques tels que le DAQPad-1200, qui se connecte dans le port parallèle de votre ordinateur, sont tous des exemples de périphériques DAQ. Les modules SCXI ne sont pas considérés comme des périphériques, à l'exception du SCXI-1200 qui est un hybride.</p>
pixel	La plus petite unité d'une image numérisée.
poignées de redimensionnement	Pointeurs en forme de chevrons au coin des objets qui indiquent les points de redimensionnement.
point	Cluster qui contient deux entiers 16 bits représentant les coordonnées horizontale et verticale.
point d'arrêt	Pause en cours d'exécution pour la mise au point.
point de coercition	Point sur un terminal indiquant celui des deux terminaux câblés ensemble que LabVIEW a converti au type de données de l'autre terminal.
pointeur	Pointe vers un pointeur sur un bloc de mémoire qui gère les tableaux et les chaînes de référence. Un tableau de chaînes est un pointeur vers un bloc de mémoire qui contient des pointeurs vers les chaînes.
polymorphisme	Capacité d'un nœud à s'ajuster automatiquement à des données de représentation, type ou structure différents.

PPC	Communication de programme à programme.
PXI	(PCI eXtensions for Instrumentation) Extensions PCI pour l'instrumentation. Plate-forme d'instrumentation modulaire basée sur un ordinateur.

R

rectangle	Cluster contenant quatre entiers 16 bits. Les deux premières valeurs décrivent les coordonnées verticale et horizontale de l'angle supérieur gauche. Les deux valeurs suivantes décrivent les coordonnées verticale et horizontale de l'angle inférieur droit.
refnum	Numéro de référence. Un identificateur que LabVIEW associe à un fichier que vous ouvrez. Utilisez le refnum pour indiquer que vous voulez qu'une fonction, ou un VI, effectue une opération sur le fichier ouvert.
refnum de fichier	<i>Voir refnum.</i>
registre à décalage	Mécanisme optionnel dans les structures en boucle utilisé pour transférer la valeur d'une variable d'une itération d'une boucle à une itération suivante. Les registres à décalage sont similaires à des variables statiques dans des langages de programmation textuels.
répertoire	Structure permettant d'organiser des fichiers en groupes. Un répertoire ressemble à une adresse qui affiche la position des fichiers. Un répertoire peut contenir des fichiers ou des sous-répertoires de fichiers.
représentation	Sous-type du type de données numériques, qui peut être représenté par des octets signés ou non signés, des mots ou des entiers longs, de même que des nombres de types flottant simple, flottant double précision ou flottant précision étendue.

S

scalaire	Nombre qu'un point sur une échelle peut représenter. Une valeur unique par opposition à un tableau. Les booléens et les clusters scalaires sont explicitement des instances singulières de leurs types de données respectifs.
----------	---

SCXI	Signal Conditioning eXtensions for Instrumentation. Gamme de produits National Instruments pour le conditionnement de signaux de bas niveau à l'intérieur d'un châssis externe près des capteurs, de telle sorte que seuls les signaux de haut niveau dans un environnement bruyant sont envoyés aux périphériques DAQ.
segment de fil de liaison	Partie de fil de liaison unique horizontale ou verticale.
situation de compétition	Survient quand au moins deux éléments de code qui s'exécutent en parallèle modifient la valeur de la même ressource partagée, habituellement une variable globale ou une variable locale.
sonde	Fonction de mise au point pour vérifier les valeurs intermédiaires d'un VI.
sortie d'erreur	Structure d'erreur qui sort d'un VI.
sous-diagramme	Diagramme à l'intérieur du cadre d'une structure.
sous-VI	VI utilisé dans le diagramme d'un autre VI. Comparable à un sous-programme.
structure	Élément de commande de programme, tel que la structure Séquence, la structure Condition, la boucle For et la boucle While.
Structure Condition	Structure de commande du branchement conditionnel, qui exécute l'un de ses sous-diagrammes, basés sur l'entrée dans la structure Condition. C'est la combinaison des déclarations IF, THEN, ELSE et CASE dans les langages de flux de la commande.
structure d'erreur	Se compose d'un indicateur d'état booléen, d'un indicateur de code numérique et d'un indicateur de source de chaîne.
structure Séquence	Structure de commande de programme qui exécute ses sous-diagrammes en ordre numérique. Utilisez cette structure pour forcer les nœuds qui ne sont pas des données dépendantes à s'exécuter dans l'ordre que vous avez défini.
symbole	Petite image ou icône.
syntaxe	Ensemble de règles auxquelles doivent se conformer les expressions dans un langage de programmation.

T

table de pixels (pixmap)	Format standard pour stocker des images dans lesquelles une valeur de couleur représente chaque pixel. Une bitmap est une version noir et blanc d'une table de pixels.
tableau	Liste ordonnée et indexée d'éléments de données du même type.
tableau vide	Tableau contenant aucun élément mais dont le type de données est défini. Par exemple, un tableau comportant une commande numérique dans sa fenêtre d'affichage des données mais n'ayant de valeurs définies pour aucun élément est un tableau numérique vide.
TCP	Transmission Control Protocol.
terminal	Objet ou zone sur un nœud par où les données sont transmises.
terminal conditionnel	Terminal de boucle While contenant une valeur booléenne qui détermine si le VI effectue une autre itération.
terminal de décompte	Terminal d'une boucle For dont la valeur détermine le nombre de fois qu'une boucle For exécute son sous-diagramme.
terminal d'itération	Terminal d'une boucle For ou d'une boucle While qui contient le nombre actuel d'itérations effectuées.
trigger externe	Impulsion en tension à partir d'une source externe qui déclenche un événement, comme une conversion analogique/numérique par exemple.
tunnel	Entrée de données ou terminal de sortie sur une structure.
type de données	Format pour informations. Dans LabVIEW, les types de données acceptables par la plupart des VIs et des fonctions sont les suivants : numérique, tableau, chaîne, booléen, chemin, refnum, énumération, waveform et cluster.

U

UDP	User Datagram Protocol.
utilitaire de configuration	Se réfère à Measurement & Automation Explorer sous Windows et à l'utilitaire de configuration NI-DAQ sur Macintosh.

V

variable globale	Accède à plusieurs VIs dans un diagramme et transfère des données entre ces VIs.
variable locale	Variable qui vous permet de lire ou d'écrire dans l'une des commandes ou dans l'un des indicateurs sur la face-avant d'un VI.
variable locale de séquence	Terminal utilisé pour transférer des données entre les cadres d'une structure Séquence.
vecteur	tableau 1D.
VI	Voir instrument virtuel (VI) .
VI actuel	VI dont la face-avant, le diagramme et l'éditeur d'icône constituent la fenêtre active.
VI brisé	VI qui ne peut pas s'exécuter parce que des erreurs se sont produites; signalé par une flèche brisée sur le bouton Exécution .
VI de niveau supérieur	VI au sommet de la hiérarchie des VIs. Ce terme fait la distinction entre le VI et ses sous-VIs.
VI Serveur	Mécanisme de contrôle des VIs et des applications de LabVIEW par programmation, localement et à distance.
Virtual Instrument Software Architecture	VISA Bibliothèque d'interfaces uniques pour le contrôle des instruments GPIB, VXI, RS-232 et d'autres types d'instruments.
VISA	Voir Virtual Instrument Software Architecture.
voie	Broche ou fil de liaison vers lequel ou à partir duquel vous appliquez le signal analogique ou numérique. Les signaux analogiques peuvent être des signaux asymétriques ou des signaux différentiels. Pour les signaux numériques, vous groupez les voies pour former des ports. Les ports se composent le plus souvent de quatre ou de huit voies numériques.
VXI	VME eXtensions for Instrumentation (bus).

Index

A

acheminement particulier. *Voir l'Aide LabVIEW.*

acquisition de données. *Voir DAQ.*

ActiveX, 18-1

accès aux applications pour lesquelles

ActiveX est activé, 18-3

appel de méthodes. *Voir l'Aide LabVIEW.*

clients, 18-3

commandes, 18-2

constantes pour définir des paramètres, 18-7

construction des sous-palettes, 3-6

conteneurs, 18-2

définition de paramètres à l'aide de
constantes, 18-7

définition des propriétés, 18-5

définition des propriétés par
programmation, 18-5

événements, 18-2

faces-avant distantes, 17-14

fonctions, 18-2

indicateurs, 18-2

insertion d'objets sur la face-avant, 18-4

interfaces personnalisées, 18-7

mise en réseau et, 17-1

Navigateur de propriétés, 18-5

nœud de propriété, 18-5

objets, 18-1

pages de propriétés, 18-5

pour les nœuds de script en cours
d'exécution, 20-2

propriétés, 18-1

sélection d'interfaces personnalisées, 18-4

serveurs, 18-6

VI Serveur, 16-1

VI, 18-2

visualisation des propriétés, 18-5

affichage

chaîne d'appelants, 6-7

éléments optionnels des objets de la
face-avant, 4-2

erreurs, 6-2

étiquettes des constantes automatiques. *Voir
l'Aide LabVIEW.*

faces-avant à distance, 17-11

info-bulles. *Voir l'Aide LabVIEW.*

mise en garde, 6-3

objets masqués de la face-avant. *Voir l'Aide
LabVIEW.*

terminaux, 5-2

aiguilles

ajout, 4-9

ajout

commandes à des bibliothèques, 3-5
graphique à l'icône du VI. *Voir l'Aide
LabVIEW.*

répertoires au chemin de recherche du VI.
Voir l'Aide LabVIEW.

terminaux vers fonctions, 5-11

VI à des bibliothèque, 3-5

alignement des objets. *Voir l'Aide LabVIEW.*

annotations, 4-18

édition. *Voir l'Aide LabVIEW.*

annulation d'erreurs existantes. *Voir l'Aide
LabVIEW.*

annuler options, 3-6

appel à distance des VIs, 16-1, 16-8

appel de code provenant de langages de
programmation textuels, 19-1

appelants

affichage, 6-7

chaîne d', 6-7

Appeler une fonction d'une DLL, 19-1

- appellation
 - commandes, 7-9
 - VIs, 7-12
- Apple events, 17-16
- Application Builder. *Reportez-vous à applications autonomes.*
- applications
 - construction autonome, 7-14
 - distribution des VIs, 7-13
 - construire un VI Serveur, 16-2
- applications autonomes
 - construction, 7-14
 - distribution des VIs, 7-13
- arbres
 - données variant, 5-19
- arithmétique. *Reportez-vous à équations.*
- attribution
 - mots de passe pour les diagrammes, 7-13
- audio, 12-8
- auto-indexation, 8-4
 - boucles For, 8-5
 - boucles While, 8-6
 - données inattendues, 6-9
- automation
 - commande refnum, 18-2
 - interfaces personnalisées, 18-7
- B**
- barre d'outils, 3-4
- barre de menus
 - masquage, 4-18
- barres de défilement
 - commandes de type menu déroulant, 4-15
 - listes déroulantes, 4-14
 - masquage, 4-18
- bibliothèque d'instruments
 - ajout de VIs et commandes, 3-5
- bibliothèque de l'utilisateur
 - ajout de VIs et commandes, 3-5
- bibliothèque PDF, 1-1
- bibliothèques
 - ajout de VIs et commandes, 3-5
 - conversion de répertoires en. *Voir l'Aide LabVIEW.*
 - conversion en répertoires. *Voir l'Aide LabVIEW.*
 - enregistrement de VIs sous, 7-11
 - emplacement suggéré pour, A-3
 - gestion, 7-12
 - instrument, A-1
 - marquage des VIs comme des VIs de niveau principal. *Voir l'Aide LabVIEW.*
 - organisation des, A-1
 - partagées, 7-14
 - distribution des VIs, 7-13
 - retrait des VIs de. *Voir l'Aide LabVIEW.*
 - structure des répertoires, A-1
 - utilisateur, A-1
 - VI, A-1
- bibliothèques partagées
 - appel à partir de LabVIEW, 19-1
 - construction, 7-14
 - distribution des VIs, 7-13
- binaire
 - création de fichiers, 13-11
 - E/S sur fichier, 13-3
 - nombres à virgule flottante, 6-8
- Boîtes de calcul, 20-2
 - illustration, 20-3
 - saisie d'équations, 20-2
 - saisie de déclarations de type langage C, 20-2
 - variables, 20-3
- boîtes de dialogue
 - commandes, 4-18
 - utilisation. *Voir l'Aide LabVIEW.*
 - commandes de type menu déroulant, 4-15
 - conception, 4-22
 - faces-avant distantes, 17-14

fichier natifs. *Voir l'Aide LabVIEW.*
 police, 4-20

boîtes de dialogue des fichiers natifs. *Voir l'Aide LabVIEW.*

boucles

- auto-indexation, 8-4
- contrôle de la synchronisation, 8-7
- données inattendues, 6-9
- For, 8-2
- infinies, 8-4
- registres à décalage, 8-6
- utilisation. *Voir l'Aide LabVIEW.*
- While, 8-3

boucles “do”. *Reportez-vous au* boucles While.

boucles “repeat-until”. *Reportez-vous au* boucles While.

boucles For, 8-2

- auto-indexation pour définir le décompte, 8-5
- contrôle de la synchronisation, 8-7
- données inattendues, 6-9
- registres à décalage, 8-6
- terminaux d'itération, 8-2
- terminaux de décompte, 8-2
- utilisation. *Voir l'Aide LabVIEW.*

boucles While, 8-3

- auto-indexation, 8-6
- contrôle de la synchronisation, 8-7
- données par défaut, 6-9
- faces-avant distantes, 17-14
- gestion d'erreur, 6-12
- infinies, 8-4
- registres à décalage, 8-6
- terminaux conditionnels, 8-3
- terminaux d'itération, 8-3
- utilisation. *Voir l'Aide LabVIEW.*

boucles While infinies, 8-4

boutons

- contrôle à l'aide des raccourcis clavier, 4-3

- face-avant, 4-11

boutons rotatifs

- Voir aussi* numériques.
- ajout de rampes de couleurs, 4-11
- face-avant, 4-9

buffers

- données variant, 5-19

buffers intelligents

- données variant, 5-19

bulles de données

- affichage pendant l'exécution en mode animation. *Voir l'Aide LabVIEW.*

C

câblage

- automatiquement, 5-13
- guides. *Voir l'Aide LabVIEW.*
- manuel, 5-11, 5-13
- outil, 5-13
- structures. *Voir l'Aide LabVIEW.*
- techniques, 5-25
- unités, 5-20

câblage automatique, 5-13

câblage manuel sur le diagramme, 5-13

cadrons

- Voir aussi* numériques.
- ajout de rampes de couleurs, 4-11
- face-avant, 4-9

calcul d'équations, 20-1

callback, 8-14

cartes instruments

- configuration, 1-4

chaîne d'appelants

- affichage, 6-7

chaînes, 9-1

- commandes

 - type de données (table), 5-3

- commandes et indicateurs, 4-11
- types d'affichage, 9-2

- comparaison, C-1
- constantes universelles, 5-5
- division. *Voir l'Aide LabVIEW.*
- édition par programme, 9-3
- fonctions, 5-8
- formatage, 9-4
 - spécificateurs, 9-4
- polymorphisme, B-5
- remplacement du texte. *Voir l'Aide LabVIEW.*
- tables, 9-3
- valeurs numériques à, 9-5
- variables globales, 10-6
- changer les vues des palettes. *Voir l'Aide LabVIEW.*
- Chemin de recherche du VI
 - édition. *Voir l'Aide LabVIEW.*
- chemins
 - ajouter des répertoires à un chemin de recherche du VI. *Voir l'Aide LabVIEW.*
 - commandes et indicateurs, 4-12
 - type de données (table), 5-3
 - utilisation. *Voir l'Aide LabVIEW.*
 - constantes universelles, 5-5
 - E/S sur fichier, 13-7
 - faces-avant distantes, 17-14
 - invalides, 4-12
 - options, 3-6
 - vides, 4-12
- chemins de répertoires. *Reportez-vous à chemins.*
- chemins invalides, 4-12
- chemins vides, 4-12
- chevauchement d'objets de la face-avant, 4-13
- CIN, 19-1
- clients
 - ActiveX, 18-3
 - LabVIEW pour les faces-avant distantes, 17-12
 - multiplés pour les faces-avant distantes, 17-11
 - Navigateur Web pour des faces-avant distantes, 17-13
 - cliquer-déposer-glisser. *Voir l'Aide LabVIEW.*
 - clonage des objets de la face-avant ou du diagramme. *Voir l'Aide LabVIEW.*
 - clusters, 9-15
 - commandes et indicateurs, 4-12
 - type de données (table), 5-3
 - comparaison, C-2
 - conversion des tableaux. *Voir l'Aide LabVIEW.*
 - déplacement. *Voir l'Aide LabVIEW.*
 - erreur, 6-11
 - composants, 6-12
 - rapports. *Voir l'Aide LabVIEW.*
 - fonctions, 5-8
 - modèles de fils de liaison, 9-15
 - ordre des éléments, 9-16
 - modification, 9-16
 - Voir aussi l'Aide LabVIEW.*
 - polymorphisme, B-6
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - code C
 - appel à partir de LabVIEW, 19-1
 - Code Interface Node, 19-1
 - code source. *Reportez-vous au diagramme.*
 - codes d'erreur définis par l'utilisateur. *Voir l'Aide LabVIEW.*
 - coloriage
 - couleurs système, 4-22
 - Voir aussi l'Aide LabVIEW.*
 - définition des couleurs utilisateur. *Voir l'Aide LabVIEW.*
 - objets d'arrière-plan. *Voir l'Aide LabVIEW.*
 - objets de la face-avant, 4-5
 - copie des couleurs. *Voir l'Aide LabVIEW.*

- objets de premier plan. *Voir l'Aide LabVIEW.*
- objets transparents. *Voir l'Aide LabVIEW.*
- commandes, 4-1
 - 2D, 4-8
 - 3D, 4-8
 - ActiveX, 18-2
 - affichage des éléments optionnels, 4-2
 - ajouter à des bibliothèques, 3-5
 - appellation, 7-9
 - boîte de couleurs, 4-10
 - Booléen, 4-11
 - utilisation. *Voir l'Aide LabVIEW.*
 - chaîne, 4-11
 - tables, 9-3
 - types d'affichage, 9-2
 - changement en indicateurs, 4-2
 - chemin, 4-12
 - utilisation. *Voir l'Aide LabVIEW.*
 - classiques, 4-8
 - cluster, 4-12
 - coloriage, 4-5
 - conception d'interfaces utilisateur, 4-21
 - configuration, 4-1
 - création dans un diagramme. *Voir l'Aide LabVIEW.*
 - définition de type, 4-1
 - dialogue, 4-18
 - utilisation. *Voir l'Aide LabVIEW.*
 - directives pour l'utilisation de la face-avant, 4-21
 - glissières, 4-9
 - groupement et verrouillage, 4-5
 - impression, 14-3
 - liste déroulante, 4-13
 - utilisation. *Voir l'Aide LabVIEW.*
 - masquage
 - Voir aussi l'Aide LabVIEW.*
 - éléments optionnels, 4-2
 - masqués. *Voir l'Aide LabVIEW.*
 - menu déroulant, 4-15
 - utilisation. *Voir l'Aide LabVIEW.*
 - nécessaires, 7-8
 - nom E/S, 4-16
 - nombre de couleurs élevé, 4-8
 - nombre de couleurs réduit, 4-8
 - numérique, 4-9
 - utilisation. *Voir l'Aide LabVIEW.*
 - numériques, 4-10
 - onglet, 4-13
 - optionnels, 7-8
 - palette, 3-1
 - navigation et recherche, 3-2
 - personnalisation, 3-4
 - raccourcis clavier, 4-3
 - rampe de couleurs, 4-10
 - redimensionnement, 4-6
 - par rapport à la taille de la fenêtre, 4-6
 - refnum, 4-17
 - utilisation. *Voir l'Aide LabVIEW.*
 - refnum automation, 18-2
 - remplacement, 4-2
 - rotatifs, 4-9
 - sous-titres pour les info-bulles des sous-VIs. *Voir l'Aide LabVIEW.*
 - tableau, 4-12
 - terminaux (table), 5-2
 - type énumération, 4-15
 - avancées, 4-16
 - utilisation. *Voir l'Aide LabVIEW.*
 - types de données (table), 5-2
- commandes de type
 - énumération, 4-15
 - avancées, 4-16
- commandes de type énumération, 4-15
 - avancées, 4-16
 - type de données (table), 5-3
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes de type liste déroulante, 4-13
 - utilisation. *Voir l'Aide LabVIEW.*

- commandes de type menu déroulant, 4-15
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes de type menu déroulant d'images, 4-15
- commandes et indicateurs 2D, 4-8
- commandes et indicateurs 3D, 4-8
- commandes et indicateurs à glissières, 4-9
 - Voir aussi* numériques.
- commandes et indicateurs booléens, 4-11
 - comparaison de valeurs, C-1
 - type de données (table), 5-3
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes et indicateurs classiques, 4-8
- commandes et indicateurs de type image
 - type de données (table), 5-4
 - utilisation, 12-1
- commandes et indicateurs numériques, 4-10
- commandes et indicateurs rotatifs, 4-9
- commandes onglet, 4-13
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes système, 17-17
- commentaire excluant une section particulière du diagramme
 - mise au point des Vis, 6-7
- communication, 17-1
 - ActiveX, 18-1
 - Apple events, 17-16
 - bas niveau, 17-15
 - canaux, 17-16
 - DataSocket, 17-2
 - DDE, 17-16
 - E/S sur fichier, 13-1
 - exécution des commandes système, 17-17
 - fonctions, 7-4
 - Macintosh, 17-16
 - PPC, 17-16
 - protocoles, 17-15
 - TCP, 17-15
 - UDP, 17-15
 - UNIX, 17-16
 - VI Serveur, 16-1
 - VI System Exec, 17-17
 - VI, 7-4
- communication de bas niveau, 17-15
- communication de programme à programme, 17-16
- communication par canaux, 17-16
- commutateurs sur la face-avant, 4-11
- compactage de mémoire. *Voir l'Aide LabVIEW.*
- comparaison
 - chaînes, C-1
 - clusters, C-2
 - numériques, C-2
 - tableaux, C-2
 - valeurs booléennes, C-1
 - versions des VIs, 7-2
- compteurs
 - Voir aussi* numériques.
 - ajout de rampes de couleurs, 4-11
 - face-avant, 4-9
- conception
 - boîtes de dialogue, 4-22
 - diagramme, 5-25
 - face-avant, 4-21
 - projets, 7-1
 - sous-VIs, 7-9
- conception du projet, 7-1
- conditions normales comme erreurs. *Voir l'Aide LabVIEW.*
- conditions par défaut, 8-8
- configuration
 - apparence et comportement des VIs, 15-1
 - menus, 15-2
 - objets de la face-avant, 4-1
 - serveurs pour les faces-avant distantes, 17-11
 - LabVIEW, 17-13
 - navigateur Web, 17-13

- connecteurs, 2-5
 - définition, 7-6
 - entrées et sorties nécessaires et optionnelles, 7-8
 - impression, 14-3
- connexion des terminaux, 5-11
- constantes, 5-4
 - création. *Voir l'Aide LabVIEW.*
 - définies par l'utilisateur, 5-5
 - définition de paramètres avec ActiveX, 18-7
 - édition. *Voir l'Aide LabVIEW.*
 - tableaux, 9-12
 - universelles, 5-5
- constantes définies par l'utilisateur, 5-5
- constantes universelles, 5-5
- construction
 - applications autonomes, 7-14
 - distribution des VIs, 7-13
 - applications des drivers d'instrument. *Voir l'Aide LabVIEW.*
 - applications du VI Serveur, 16-2
 - bibliothèques partagées, 7-14
 - distribution des VIs, 7-13
 - diagramme, 5-1
 - face-avant, 4-1
 - sous-VIs, 7-4
 - VIs, 7-1
 - VIs polymorphes, 5-16
- conteneurs
 - ActiveX, 18-2
- contrôle
 - code source, 7-2
 - instruments, 7-3
 - objets de la face-avant à distance. *Voir l'Aide LabVIEW.*
 - objets de la face-avant par programmation, 16-9
 - par programmation des VIs, 16-1
 - VIs à distance, 17-11
 - VIs lorsqu'ils sont appelés comme sous-VIs, 7-3
- contrôle de code source, 7-2
- conventions utilisées dans ce manuel, xx
- conversion
 - bibliothèques en répertoires. *Voir l'Aide LabVIEW.*
 - de tableaux en clusters et vice-versa. *Voir l'Aide LabVIEW.*
 - répertoires en bibliothèques. *Voir l'Aide LabVIEW.*
 - types de données LabVIEW en HTML, 9-7
 - valeurs numériques en chaînes, 9-5
 - XML en données LabVIEW, 9-8
- copie
 - objets sur la face-avant ou le diagramme. *Voir l'Aide LabVIEW.*
 - VIs, A-3
- correction
 - VIs, 6-2
 - techniques de mise au point, 6-4
- couleur
 - boîtes, 4-10
 - commandes et indicateurs utilisant un nombre de couleurs élevé, 4-8
 - commandes et indicateurs utilisant un nombre de couleurs réduit, 4-8
 - création dans des graphiques, 12-6
 - modification dans des graphiques, 12-6
 - options, 3-6
 - rampes, 4-10
 - commandes et indicateurs rotatifs, 4-11
 - représentation, 11-13
 - sélecteur, 4-10
 - couleur d'arrière-plan des objets de la face-avant. *Voir l'Aide LabVIEW.*
 - couleur de premier plan des objets de la face-avant. *Voir l'Aide LabVIEW.*

couleurs définies par l'utilisateur. *Voir l'Aide LabVIEW.*

couleurs symboliques
Reportez-vous à couleurs système.

couleurs système, 4-22

couleurs système. *Voir l'Aide LabVIEW.*

courbes
ajout à des graphes et graphes déroulants.
Voir l'Aide LabVIEW.

empilées, 11-7

lissage, 11-2

superposées, 11-7

courbes empilées, 11-7

courbes lissées, 11-2

courbes Smith, 12-4

courbes superposées, 11-7

création
constantes définies par l'utilisateur, 5-5

descriptions des objets, 14-2

descriptions des VIs, 14-2

fichiers binaires, 13-11

fichiers journaux, 13-11

fichiers tableur, 13-9

fichiers texte, 13-9

graphes déroulants. *Voir l'Aide LabVIEW.*

graphes. *Voir l'Aide LabVIEW.*

historique des révisions, 14-2

icônes, 7-8

info-bulles, 14-2

menus, 15-2

références des commandes. *Voir l'Aide LabVIEW.*

sous-palettes. *Voir l'Aide LabVIEW.*

sous-VIs, 7-9
situations à éviter. *Voir l'Aide LabVIEW.*

tableaux, 9-12

vues de palettes, 3-5

curseurs
ajout aux graphes. *Voir l'Aide LabVIEW.*

graphe, 11-4

D

DAQ

Assistant Entrées/Sorties, 1-4

Assistant Solutions, 1-4

transmission de noms de voie, 4-16

Utilitaire de configuration, 1-4

VIs et fonctions, 7-3

DataSocket, 17-2

contrôle des objets de la face-avant. *Voir l'Aide LabVIEW.*

diagramme, 17-7

données variant, 17-8

face-avant, 17-5

pour les données, 17-5

protocoles, 17-3

URL, 17-3

DDE, 17-16

déclarations. *Reportez-vous à nœuds.*

défilement dans un diagramme, 11-3

définition de type, 4-1

définition.

codes d'erreur. *Voir l'Aide LabVIEW.*

couleurs utilisateur. *Voir l'Aide LabVIEW.*

dégrouper des objets de la face-avant, 4-5

dépannage. *Reportez-vous à mise au point.*

dépassement des limites inférieures pour les nombres, 6-8

dépassement des limites supérieures, 6-8

dépendance de données artificielle, 5-23

dépendance des données, 5-22

absente, 5-23

artificielle, 5-23

contrôle avec des structures
Séquence, 8-10

paramètres dupliqués, 13-14

situations de compétition, 10-5

déplacement

clusters. *Voir l'Aide LabVIEW.*

fils de liaison. *Voir l'Aide LabVIEW.*

objets. *Voir l'Aide LabVIEW.*

- sous-palettes. *Voir l'Aide LabVIEW.*
 - tableaux. *Voir l'Aide LabVIEW.*
- dernier entré, premier sorti
 - données variant, 5-19
- désactivation
 - outils de mise au point, 6-8
 - sections d'un diagramme
 - mise au point des VIs, 6-7
- développement de VIs, 7-1
 - directives, 1-3
 - suivi du développement. *Reportez-vous à documentation des VIs.*
- déverrouillage
 - objets de la face-avant, 4-5
 - VIs. *Voir l'Aide LabVIEW.*
- diagramme, 2-3
 - ajout d'espace sans redimensionner, 5-26
 - alignement des objets. *Voir l'Aide LabVIEW.*
 - cablage automatique, 5-13
 - câblage manuel, 5-11, 5-13
 - commentaire excluant une section particulière, 6-7
 - conception, 5-25
 - constantes, 5-4
 - contrôler le code source, 7-2
 - copie d'objets. *Voir l'Aide LabVIEW.*
 - création de commandes et d'indicateurs. *Voir l'Aide LabVIEW.*
 - DataSocket, 17-7
 - données variant, 5-18
 - espacement d'objets de façon régulière. *Voir l'Aide LabVIEW.*
 - étiquettes, 4-18
 - création. *Voir l'Aide LabVIEW.*
 - édition. *Voir l'Aide LabVIEW.*
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - fonctions, 5-7
 - impression, 14-6
 - insertion d'objets. *Voir l'Aide LabVIEW.*
 - nœuds, 5-6
 - objets, 5-1
 - options, 3-6
 - planification, 7-1
 - points de coercition, 5-14
 - polices, 4-19
 - protection par mot de passe, 7-13
 - recherche de terminaux. *Voir l'Aide LabVIEW.*
 - remplacement des objets. *Voir l'Aide LabVIEW.*
 - réorganisation d'objets. *Voir l'Aide LabVIEW.*
 - structures, 8-1
 - utilisation. *Voir l'Aide LabVIEW.*
 - suppression d'objets. *Voir l'Aide LabVIEW.*
 - terminaux, 5-2
 - affichage, 5-2
 - ajout de fonctions, 5-11
 - commandes et indicateurs (table), 5-2
 - objets de la face-avant et, 5-1
 - suppressions en provenance de fonctions, 5-11
 - types de données (table), 5-2
 - VI Serveur, 16-1
 - dimensionnement. *Reportez-vous à redimensuinement.*
 - dimensions
 - tableaux, 9-9
 - directives de développement, 1-3
 - distribution
 - des objets de la face-avant. *Voir l'Aide LabVIEW.*
 - VIs, 7-13
 - division
 - chaînes. *Voir l'Aide LabVIEW.*
 - DLL
 - appel à partir de LabVIEW, 19-1
 - construction, 7-14
 - distribution des VIs, 7-13

- documentation
 - bibliothèque PDF, 1-1
 - conventions utilisées dans ce manuel, xx
 - guide, 1-1
 - introduction à ce manuel, xix
 - organisation de ce manuel, xx
 - structure des répertoires, A-2
 - utilisation avec d'autres ressources, 1-1
 - utilisation de ce manuel, xix
 - documentation des VIs, 14-1
 - création d'info-bulles, 14-2
 - création de descriptions des VIs et objets, 14-2
 - fichiers d'aide, 14-5
 - historique des révisions, 14-2
 - impression, 14-3
 - lien vers des fichiers d'aide que vous créez. *Voir l'Aide LabVIEW.*
 - par programmation, 14-4
 - données aplaties
 - données variant et, 5-19
 - données bufférisées
 - variables locales, 10-6
 - données indéfinies, 6-8
 - boucles For, 6-9
 - boucles While, 6-9
 - éviter, 6-10
 - Inf (infinity), 6-8
 - NaN (not a number), 6-8
 - recherche, 6-8
 - tableaux, 6-9
 - données par défaut
 - boucles While, 6-9
 - tableaux, 6-9
 - données variant, 5-18
 - ActiveX, 18-2
 - commandes et indicateurs
 - type de données (table), 5-3
 - conversion en, 5-18
 - DataSocket, 17-8
 - données aplaties et, 5-19
 - édition des attributs. *Voir l'Aide LabVIEW.*
 - gestion, 5-18
 - données waveform
 - commandes et indicateurs
 - type de données (table), 5-3
 - fonctions, 5-10
 - graphes, 11-8
 - graphiques, 12-3
 - types de données, 11-8, 11-9
 - graphes déroulants, 11-11
 - type de données, 11-19
 - drivers
 - instrument
 - LabVIEW. *Voir l'Aide LabVIEW.*
 - drivers d'instrument
 - LabVIEW. *Voir l'Aide LabVIEW.*
 - Dynamic Data Exchange, 17-16
- ## E
- E/S
 - commandes et indicateurs
 - type de données (table), 5-4
 - commandes et indicateurs de nom E/S, 4-16
 - erreur, 6-11
 - fichier. *Reportez-vous à E/S sur fichier.*
 - E/S sur fichier, 13-1
 - chemins, 13-7
 - écriture de waveforms, 13-12
 - enregistrement de données de la face-avant, 13-17
 - enregistrement sur disque en continu, 13-8
 - fichiers binaires, 13-3
 - création, 13-11
 - fichiers journaux, 13-4
 - création, 13-11
 - fichiers standard, 13-4

- fichiers tableur
 - création, 13-9
- fichiers texte, 13-2
 - création, 13-9
- fonctions, 5-9
- fonctions de fichiers avancées, 13-7
- formats, 13-2
- lecture de waveforms, 13-13
- mise en réseau et, 17-1
- opération de base, 13-1
- paramètres dupliqués, 13-14
- refnums, 13-1
- VIs de fichiers de configuration
 - format, 13-15
 - lecture et écriture de fichiers
 - .ini, 13-14
 - spécial, 13-15
- VIs de haut niveau, 13-6
- VIs et fonctions de bas niveau, 13-7
- E/S sur fichier journal, 13-4
 - création de fichiers, 13-11
- échelles x
 - formatage, 11-5
 - multiples, 11-2
- échelles y
 - formatage, 11-5
 - multiples, 11-2
- écrire dans des fichiers, 13-1
- Éditeur de menus, 15-2
- édition
 - étiquettes. *Voir l'Aide LabVIEW.*
 - menus, 15-2
 - menus locaux des VIs polymorphes. *Voir l'Aide LabVIEW.*
 - objets de la face-avant, 4-1
 - vues de palettes, 3-5
- effacement
 - des enregistrements de données, 13-19
 - éléments de tableau. *Voir l'Aide LabVIEW.*
 - fils de liaison brisés, 5-14
 - graphes et graphes déroulants. *Voir l'Aide LabVIEW.*
 - indicateurs. *Voir l'Aide LabVIEW.*
 - objets sur la face-avant ou le diagramme. *Voir l'Aide LabVIEW.*
 - structures. *Voir l'Aide LabVIEW.*
 - vues de palettes. *Voir l'Aide LabVIEW.*
- éliminer des enregistrements de données, 13-19
- emplacement pour enregistrer des fichiers, A-3
- enregistrement
 - options de l'environnement de travail, 3-7
- enregistrement de données. *Reportez-vous à journal.*
- enregistrement de fichiers
 - emplacement suggéré pour, A-3
- enregistrement des VIs, 7-10
 - au format antérieur, 7-12
 - bibliothèques, 7-11
 - fichiers individuels, 7-11
 - revenir aux dernières versions enregistrées. *Voir l'Aide LabVIEW.*
- enregistrement sur disque en continu, 13-8
- enregistrements, 13-17
 - effacement, 13-19
 - spécification pendant la récupération des données de la face-avant au moyen des sous-VIs, 13-21
- entiers
 - conversion, B-1
 - dépassement des limites inférieures et supérieures, 6-8
- équations
 - Boîtes de calcul, 20-2
 - HiQ
 - ActiveX, 20-2
 - nœud de script, 20-5
 - scripts de mise au point, 20-7
 - VIs, 20-5
 - intégration dans LabVIEW, 20-1

- MATLAB
 - ActiveX, 20-2
 - nœud de script, 20-5
 - scripts de mise au point, 20-7
 - méthodes d'utilisation, 20-1
 - nœuds d'expression, 20-4
- erreurs
 - acheminement particulier. *Voir l'Aide LabVIEW.*
 - affichage, 6-2
 - annulation d'erreurs existantes. *Voir l'Aide LabVIEW.*
 - clusters, 6-11
 - composants, 6-12
 - connecteur, 7-7
 - rapports. *Voir l'Aide LabVIEW.*
 - codes, 6-11
 - conditions normales comme. *Voir l'Aide LabVIEW.*
 - définition d'erreurs personnalisées. *Voir l'Aide LabVIEW.*
 - E/S, 6-11
 - fenêtre, 6-2
 - gestion, 6-10
 - commande d'instrument. *Voir l'Aide LabVIEW.*
 - méthodes, 6-11
 - utilisation des boucles While, 6-12
 - utilisation des structures
 - Condition, 6-12
 - liste, 6-2
 - notification. *Voir l'Aide LabVIEW.*
 - recherche, 6-2, 6-10
 - techniques de mise au point, 6-4
 - unités incompatibles, 5-20
 - VIs brisés, 6-2
- espace de travail
 - ajout à la face-avant ou au diagramme, 4-8
- espace disque
 - options, 3-6
 - vérification. *Voir l'Aide LabVIEW.*
- espacement d'objets de façon régulière. *Voir l'Aide LabVIEW.*
- étiquetage, 4-18
 - constantes, 5-4
 - création d'étiquettes libres. *Voir l'Aide LabVIEW.*
 - édition. *Voir l'Aide LabVIEW.*
 - polices, 4-19
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - sous-titres, 4-19
 - unités de mesures, 5-19
 - variables globales, 10-3
 - variables locales, 10-2
- étiquettes
 - affichage des constantes automatiques. *Voir l'Aide LabVIEW.*
 - transparents. *Voir l'Aide LabVIEW.*
- étiquettes d'unité, 5-19
- étiquettes dépendantes, 4-18
 - édition. *Voir l'Aide LabVIEW.*
- étiquettes des constantes automatiques
 - affichage. *Voir l'Aide LabVIEW.*
- étiquettes libres, 4-18
 - création. *Voir l'Aide LabVIEW.*
- événements
 - ActiveX, 18-2
 - disponibles dans LabVIEW. *Voir l'Aide LabVIEW.*
 - filtre, 8-15
 - gestion, 8-14
 - avertissements. *Voir l'Aide LabVIEW.*
 - notification, 8-15
- exécution
 - flux
 - contrôle avec des structures
 - Séquence, 8-10
 - mise en surbrillance
 - afficher les bulles de donnée. *Voir l'Aide LabVIEW.*
 - mise au point des VIs, 6-4

- sondage automatique. *Voir l'Aide LabVIEW.*
- suspension
 - mise au point des VIs, 6-6
- exécution des commandes système, 17-17
- exécution des VIs, 6-1
- exécution des VIs par incrémentation, 6-5
- exécution en continu des VIs, 6-1
- exemples, 1-4
 - tableaux, 9-9
 - tableaux 1D, 9-9
 - tableaux 2D, 9-11

F

- face-avant, 2-2
 - affichage avec différentes résolutions d'écran, 4-22
 - affichage des éléments optionnels des objets, 4-2
 - ajout d'espace sans redimensionner, 4-8
 - alignement des objets. *Voir l'Aide LabVIEW.*
 - caractéristiques du texte, 4-19
 - changement de commandes en indicateurs et vice versa, 4-2
 - chevauchement d'objets, 4-13
 - coloriage des objets, 4-5
 - arrière-plan et premier plan. *Voir l'Aide LabVIEW.*
 - copie des couleurs. *Voir l'Aide LabVIEW.*
 - commandes, 4-8
 - conception, 4-21
 - configuration des objets, 4-1
 - contrôle à distance, 17-11
 - contrôle d'objets à distance. *Voir l'Aide LabVIEW.*
 - contrôler les objets par programmation, 16-9
 - copie d'objets. *Voir l'Aide LabVIEW.*
 - DataSocket, 17-5
 - définition de la taille de la fenêtre. *Voir l'Aide LabVIEW.*
 - définition de type, 4-1
 - distribution d'objets. *Voir l'Aide LabVIEW.*
 - effacement des indicateurs. *Voir l'Aide LabVIEW.*
 - enregistrement de données, 13-17
 - espacement d'objets de façon régulière. *Voir l'Aide LabVIEW.*
 - étiquettes, 4-18
 - création. *Voir l'Aide LabVIEW.*
 - édition. *Voir l'Aide LabVIEW.*
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - groupement et verrouillage des objets, 4-5
 - importation de graphiques, 4-5
 - impression, 14-6
 - indicateurs, 4-8
 - insertion d'objets à l'aide d'ActiveX, 18-4
 - journal, 13-17
 - masquage
 - éléments optionnels des objets, 4-2
 - objets. *Voir l'Aide LabVIEW.*
 - mise à l'échelle des objets, 4-6
 - objets
 - terminaux du diagramme et, 5-1
 - objets masqués. *Voir l'Aide LabVIEW.*
 - objets transparents. *Voir l'Aide LabVIEW.*
 - options, 3-6
 - ordre des objets, 4-4
 - paramétrage de l'ordre de navigation, 4-4
 - planification, 7-1
 - polices, 4-19
 - publication d'images sur le Web, 17-10
 - raccourcis clavier, 4-3
 - recherche d'objets. *Voir l'Aide LabVIEW.*
 - récupération des données
 - utilisation de sous-VIs, 13-20

- utilisation des fonctions d'E/S sur fichier, 13-22
- redimensionnement des objets, 4-6
 - par rapport à la taille de la fenêtre, 4-6
- remplacement des objets, 4-2
- réorganisation d'objets. *Voir l'Aide LabVIEW.*
- sous-titres, 4-19
 - création. *Voir l'Aide LabVIEW.*
- sous-VIs, 7-9
- suppression d'objets. *Voir l'Aide LabVIEW.*
- visualisation à distance, 17-11
- faire glisser et déposer. *Voir l'Aide LabVIEW.*
- Fenêtre de hiérarchie, 7-10
 - impression, 14-3
 - recherche. *Voir l'Aide LabVIEW.*
- fichier de menu d'exécution, 15-2
- fichier rtm, 15-2
- fichier serpdrv, A-2
- fichiers bitmap, 12-7
- fichiers BMP, 12-7
- fichiers d'aide, 1-2
 - création de vos propres, 14-5
 - HTML, 14-5
 - lien vers des VIs. *Voir l'Aide LabVIEW.*
 - RTF, 14-5
- fichiers de support
 - pour appeler HiQ, 20-8
- fichiers GIF, 14-4
- fichiers ini
 - lecture et écriture, 13-14
- fichiers Joint Photographic Expert Group, 12-7
- fichiers JPEG, 12-7, 14-4
 - Serveur Web, 17-10
- fichiers PNG, 12-7, 14-4
 - Serveur Web, 17-11
- fichiers Portable Network Graphics, 12-7
- fichiers standard, 13-4
- fichiers tableur
 - création, 13-9
 - écriture de données numériques sur des, 9-5
- fichiers texte
 - création, 13-9
 - E/S sur fichier, 13-2
 - écriture de données numériques sur des, 9-5
- FIFO
 - données variant, 5-19
- files d'attente
 - données variant, 5-19
- fils de liaison, 2-4
 - brisés, 5-14
 - déplacement. *Voir l'Aide LabVIEW.*
 - sélection, 5-14
- fils de liaison brisés, 5-14
- flux de données
 - observation, 6-4
- fonctions, 5-7
 - ajouts de terminaux, 5-11
 - avancées, 5-10
 - Booléen, 5-7
 - chaîne, 5-8
 - cluster, 5-8
 - contrôle d'applications, 5-10
 - diagramme, 5-7
 - dialogue, 5-9
 - données waveform, 5-10
 - E/S sur fichier, 5-9
 - numérique, 5-7
 - palette, 3-1
 - navigation et recherche, 3-2
 - personnalisation, 3-4
 - titre des fenêtres. *Voir l'Aide LabVIEW.*
 - polymorphes, B-1
 - recherche. *Voir l'Aide LabVIEW.*
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - référence. *Voir l'Aide LabVIEW.*

- suppression de terminaux, 5-11
- tableau, 5-8
- temps, 5-9
- fonctions avancées, 5-10
- Fonctions booléennes, 5-7
 - polymorphisme, B-4
- Fonctions de comparaison, C-1
 - polymorphisme, B-6
- fonctions de contrôle d'applications, 5-10
- fonctions dialogue, 5-9
- fonctions logarithmiques
 - polymorphisme, B-8
- fonctions temps, 5-9
- format de l'heure
 - options, 3-7
- format de la date
 - options, 3-7
- formatage
 - chaînes, 9-4
 - spécificateurs, 9-4
 - échelles d'un graphe, 11-5
 - texte sur la face-avant, 4-19
- formatage des caractères, 4-19
- formation, E-1
- formats pour l'E/S sur fichier, 13-2
 - fichiers binaires, 13-3
 - fichiers journaux, 13-4
 - fichiers texte, 13-2
- formules. *Reportez-vous à équations.*

G

- génération de rapports, 14-6
 - clusters d'erreur. *Voir l'Aide LabVIEW.*
- glissières
 - ajout, 4-9
- GPIB
 - configuration, 1-4
- graphe à balayage, 11-7

- graphe d'intensité, 11-12
 - options, 11-15
 - représentation des couleurs, 11-13
- graphe déroulant, 11-7
- graphes, 11-1
 - 3D, 11-17
 - ajout de tracés. *Voir l'Aide LabVIEW.*
 - courbes lissées, 11-2
 - courbes Smith, 12-4
 - création. *Voir l'Aide LabVIEW.*
 - curseurs, 11-4
 - ajout. *Voir l'Aide LabVIEW.*
 - données waveform, 11-8
 - types de données, 11-8, 11-9
 - échelles
 - formatage, 11-5
 - échelles multiples, 11-2
 - effacement. *Voir l'Aide LabVIEW.*
 - graphiques, 12-3
 - intensité, 11-12
 - options, 11-15
 - lignes de transmission, 12-4
 - mise à l'échelle, 11-5
 - numériques, 11-15
 - masquage des données, 11-17
 - options, 11-2
 - personnalisation de l'apparence, 11-3
 - personnalisation du comportement, 11-4
 - polaires, 12-3
 - rafraîchissements progressifs, 11-6
 - types, 11-1
 - XY, 11-8
 - types de données, 11-10, 11-11
 - zoom. *Voir l'Aide LabVIEW.*
- graphes 3D, 11-17
- graphes déroulants, 11-1
 - ajout de tracés. *Voir l'Aide LabVIEW.*
 - courbes empilées, 11-7
 - courbes lissées, 11-2
 - courbes superposées, 11-7

création. *Voir l'Aide LabVIEW.*
défilement, 11-3
données waveform, 11-11
échelles multiples, 11-2
effacement. *Voir l'Aide LabVIEW.*
faces-avant distantes, 17-14
intensité, 11-12
 options, 11-14
longueur de l'historique, 11-6
options, 11-2
personnalisation de l'apparence, 11-3
personnalisation du comportement, 11-6
types, 11-1
zoom. *Voir l'Aide LabVIEW.*
graphes déroulants d'intensité, 11-12
 options, 11-14
 représentation des couleurs, 11-13
graphes numériques, 11-15
 courbes lissées, 11-2
 masquage des données, 11-17, D-1
graphes polaires, 12-3
graphes XY, 11-8
 types de données, 11-10, 11-11
graphiques
 ajout à l'icône du VI. *Voir l'Aide LabVIEW.*
 commandes et indicateurs de type image
 type de données (table), 5-4
 utilisation, 12-1
 création de couleurs, 12-6
 entrée de texte, 12-5
 faire glisser et déposer. *Voir l'Aide LabVIEW.*
 formats, 12-7
 pour fichiers HTML, 14-4
 graphes, 12-3
 importation, 4-5
 modification de couleurs, 12-6
 publication de faces-avant sur le Web, 17-10

 tables de pixels, 12-5
 tracé de formes, 12-5
groupement
 données
 chaînes, 9-1
 clusters, 9-15
 tableaux, 9-9
 objets de la face-avant, 4-5
 VIs dans des bibliothèques, 7-11

H

HiQ

ActiveX, 20-2
fichiers support pour applications
 LabVIEW, 20-8
lancement à partir de LabVIEW, 20-5
nœud de script, 20-5
scripts de mise au point, 20-7
types de données, 20-6
VIs, 20-5

historique

Reportez-vous aussi à historique des révisions.
graphes déroulants, 11-6
options, 3-7
historique des révisions
 création, 14-2
 impression, 14-3
 nombres, 14-2

HTML

Reportez-vous aussi à Web.
création de documents, 17-10
enregistrement dans un fichier, 14-3
fichiers d'aide, 14-5
formats graphiques, 14-4
Impression de rapports, 14-6
publication de VIs sur le Web, 17-9

I

- icônes, 2-5
 - création, 7-8
 - édition, 7-8
 - impression, 14-3
- ignorer les touches de fonction par défaut. *Voir l'Aide LabVIEW.*
- images animées des faces-avant, 17-10
- images statiques de faces-avant, 17-10
- images. *Reportez-vous à graphiques.*
- impédance des lignes de transmission, 12-4
- importation de graphiques, 4-5
- impression, 14-5
 - à la fin de l'exécution, 14-7
 - documentation des VIs, 14-3
 - enregistrement
 - dans un fichier HTML, 14-3
 - dans un fichier RTF, 14-3
 - fenêtre active, 14-6
 - options, 3-7
 - par programmation, 14-7
 - rapports, 14-6
 - techniques, 14-8
 - utilisation de sous-VIs, 14-8
- impression PostScript. *Voir l'Aide LabVIEW.*
- indexation des boucles, 8-4
 - boucles For, 8-5
 - boucles While, 8-6
- indicateurs, 4-1
 - 2D, 4-8
 - 3D, 4-8
 - ActiveX, 18-2
 - affichage des éléments optionnels, 4-2
 - boîte de couleur, 4-10
 - Booléen, 4-11
 - utilisation. *Voir l'Aide LabVIEW.*
 - chaîne, 4-11
 - types d'affichage, 9-2
 - changement en commandes, 4-2
 - chemin, 4-12
 - utilisation. *Voir l'Aide LabVIEW.*
 - classiques, 4-8
 - cluster, 4-12
 - coloriage, 4-5
 - conception d'interfaces utilisateur, 4-21
 - configuration, 4-1
 - création dans un diagramme. *Voir l'Aide LabVIEW.*
 - définition de type, 4-1
 - directives pour l'utilisation de la face-avant, 4-21
 - effacement. *Voir l'Aide LabVIEW.*
 - glissières, 4-9
 - groupement et verrouillage, 4-5
 - impression, 14-3
 - masquage
 - Voir aussi l'Aide LabVIEW.*
 - éléments optionnels, 4-2
 - masqués. *Voir l'Aide LabVIEW.*
 - nécessaires, 7-8
 - nom E/S, 4-16
 - nombre de couleurs élevé, 4-8
 - nombre de couleurs réduit, 4-8
 - numérique, 4-9
 - utilisation. *Voir l'Aide LabVIEW.*
 - numériques, 4-10
 - onglet, 4-13
 - optionnels, 7-8
 - rampe de couleurs, 4-10
 - redimensionnement, 4-6
 - par rapport à la taille de la fenêtre, 4-6
 - refnum, 4-17
 - utilisation. *Voir l'Aide LabVIEW.*
 - remplacement, 4-2
 - rotatifs, 4-9
 - tableau, 4-12
 - terminaux (table), 5-2
 - type énumération
 - avancées, 4-16
 - types de données (table), 5-2

indices des tableaux, 9-9
 affichage, 9-13
Inf (infinity) valeur flottante
 données indéfinies, 6-8
info-bulles
 affichage sur les terminaux. *Voir l'Aide LabVIEW.*
 affichage. *Voir l'Aide LabVIEW.*
 création, 14-2
 sous-titres des commandes. *Voir l'Aide LabVIEW.*
initiation, 1-1
insertion
 d'objets dans des palettes. *Voir l'Aide LabVIEW.*
 éléments dans des tableaux. *Voir l'Aide LabVIEW.*
 objets sur le diagramme. *Voir l'Aide LabVIEW.*
installateurs
 construction, 7-14
instances de sous-VIs
 détermination, 6-7
 suspension de l'exécution, 6-6
instruments
 configuration, 1-4
 contrôle, 7-3
instruments virtuels. *Reportez-vous aux VIs.*
intégration d'objets à l'aide d'ActiveX, 18-4
intégration de systèmes, E-2
interface utilisateur. *Reportez-vous à la face-avant.*
interfaces automation personnalisées, 18-7
Internet. *Reportez-vous à Web.*
IVI
 drivers d'instrument. *Voir l'Aide LabVIEW.*
 transmission de noms logiques, 4-16

J

jauges
 Voir aussi numériques.
 ajout de rampes de couleurs, 4-11
 face-avant, 4-9
journal, 13-17
 automatique, 13-17
 changer la liaison vers le fichier journal, 13-19
 effacer des enregistrements, 13-19
 effacer la liaison vers le fichier journal, 13-19
 extraction des données par programmation, 13-20
 interactif, 13-18

L

LabVIEW, 1-1
 options, 3-6
 personnalisation, 3-6
labview.ini, 3-7
lancement de VIs à partir de la ligne de commande. *Voir l'Aide LabVIEW.*
lecture de fichiers, 13-1
liaison de VIs vers des fichiers d'aide HTML ou compilés. *Voir l'Aide LabVIEW.*
liaison vers le fichier journal, 13-17
 changer, 13-19
 effacement, 13-19
libération de la mémoire. *Voir l'Aide LabVIEW.*
licences pour servir des faces-avant distantes, 17-12
LIFO
 données variant, 5-19
ligne de commande
 lancement de VIs. *Voir l'Aide LabVIEW.*
lignes de transmission, 12-4

liste de paramètres. *Reportez-vous à connecteurs.*
 liste déroulante à colonnes multiples. *Reportez-vous à commandes.*
 liste. *Reportez-vous à affichage.*
 localisation des VIs, 7-13
 login automatique. *Voir l'Aide LabVIEW.*

M

manuel. *Reportez-vous à la documentation.*
 marquage des VIs comme des VIs de niveau principal dans des bibliothèques. *Voir l'Aide LabVIEW.*
 masquage
 barre de menus, 4-18
 barres de défilement, 4-18
 éléments optionnels des objets de la face-avant, 4-2
 objets de la face-avant. *Voir l'Aide LabVIEW.*
 masquage des données numériques, 11-17, D-1
 mathématiques. *Reportez-vous à équations.*
 MATLAB
 ActiveX, 20-2
 nœud de script, 20-5
 scripts de mise au point, 20-7
 types de données, 20-6
 Measurement & Automation Explorer, 1-4
 mémoire
 compactage. *Voir l'Aide LabVIEW.*
 désactivation des outils de mise au point, 6-8
 gestion avec un modèle de programmation par flux de données, 5-24
 lecture et écriture avec des données variant, 5-19
 libération. *Voir l'Aide LabVIEW.*
 points de coercition, 5-14
 variables globales, 10-6
 variables locales, 10-6
 menus, 3-3
 commandes de type menu déroulant, 4-15
 directs. *Voir l'Aide LabVIEW.*
 édition, 15-2
 gestion des sélections, 15-3
 locaux, 3-3
 édition pour les VIs polymorphes. *Voir l'Aide LabVIEW.*
 référence. *Voir l'Aide LabVIEW.*
 menus déroulants sur la face-avant, 4-15
 menus directs. *Voir l'Aide LabVIEW.*
 menus locaux, 3-3
 en mode exécution, 3-4
 menus locaux. *Reportez-vous aux menus locaux.*
 message de login au démarrage affichage. *Voir l'Aide LabVIEW.*
 méthodes
 ActiveX, 18-1
 mis en réseau. *Reportez-vous à communication.*
 mise à jour des palettes. *Voir l'Aide LabVIEW.*
 mise à jour des VIs, 7-12
 mise à l'échelle
 graphes, 11-5
 objets de la face-avant, 4-6
 mise au point
 boucles, 6-9
 désactivation des outils de mise au point, 6-8
 données indéfinies, 6-8
 données par défaut, 6-9
 fils de liaison masqués, 5-25
 options, 3-6
 outils
 désactivation, 6-8
 scripts HiQ et MATLAB, 20-7
 structures. *Voir l'Aide LabVIEW.*

- techniques, 6-4
 - commentaire excluant une section particulière du diagramme, 6-7
 - gestion d'erreur, 6-10
 - mode Animation, 6-4
 - mode pas à pas, 6-5
 - outil Point d'arrêt, 6-6
 - outil Sonde, 6-5
 - suspension de l'exécution, 6-6
- VIs brisés, 6-2
- VIs exécutables. *Voir l'Aide LabVIEW.*
- mise en surbrillance de l'exécution
 - mise au point des VIs, 6-4
- mises en garde
 - affichage, 6-3
 - affichage par défaut. *Voir l'Aide LabVIEW.*
 - bouton, 6-3
- mode exécution
 - ouverture des VIs dans. *Voir l'Aide LabVIEW.*
- mode pas à pas
 - mise au point des VIs, 6-5
- modèle de programmation par flux de données
 - gestion de mémoire, 5-24
- modèle de programmation séquentielle, 5-22
- motifs
 - terminal, 7-7

N

- NaN (not a number) valeur flottante
 - données indéfinies, 6-8
- National Instruments support sur le Web, E-1
- NI Developer Zone, E-1
- niveau de coopération
 - paramétrage. *Voir l'Aide LabVIEW.*
- nœud d'appel par référence, 16-8
- nœud de méthode, 16-5
 - ActiveX, 18-2
- nœud de propriété, 16-4
 - ActiveX, 18-5
 - modification des éléments de la liste déroulante, 4-14
 - recherche d'objets ou de terminaux. *Voir l'Aide LabVIEW.*
- nœuds, 2-4
 - appel par référence, 16-8
 - appeler, 16-5
 - diagramme, 5-6
 - flux d'exécution, 5-22
 - nœud de script HiQ, 20-5
 - nœud de script MATLAB, 20-5
 - Propriété, 16-4
 - redimensionnement. *Voir l'Aide LabVIEW.*
- nœuds d'expression, 20-4
- nœuds de script
 - HiQ, 20-5
 - MATLAB, 20-5
- nombre
 - dépassement des limites inférieures et supérieures, 6-8
- nombre à virgule flottante
 - conversion, B-1
 - dépassement des limites inférieures et supérieures, 6-8
- nombre hors gamme, 4-16
- not a number (NaN) valeur flottante
 - données indéfinies, 6-8
- notes d'application, 1-3
- notification d'erreurs. *Voir l'Aide LabVIEW.*
- numériques
 - chaînes et, 9-5
 - changement de la représentation. *Voir l'Aide LabVIEW.*
 - commandes et indicateurs, 4-9
 - utilisation. *Voir l'Aide LabVIEW.*
 - comparaison, C-2
 - constantes universelles, 5-5
 - conversion, B-1

écriture de données sur des fichiers texte
ou tableur, 9-5
équations, 20-1
fonctions, 5-7
formules, 20-1
hors gamme, 4-16
polymorphisme, B-2
types de données (table), 5-2
unités de mesures, 5-19
numéro de révision
affichage dans la barre de titre. *Voir l'Aide
LabVIEW.*

O

Objet d'application
manipulation des paramètres, 16-4
VI Serveur, 16-3
Objet du VI
manipulation des paramètres, 16-4
VI Serveur, 16-3
objets
ActiveX, 18-1
affichage des éléments optionnels, 4-2
alignement. *Voir l'Aide LabVIEW.*
câblage automatique sur le
diagramme, 5-13
câblage manuel sur le diagramme, 5-11
changement de commandes en indicateurs
et vice versa, 4-2
chevauchement sur la face-avant, 4-13
coloriage sur la face-avant, 4-5
copie des couleurs. *Voir l'Aide
LabVIEW.*
configuration sur la face-avant, 4-1
contrôle par programmation, 16-9
création d'info-bulles, 14-2
création de descriptions, 14-2
déplacement. *Voir l'Aide LabVIEW.*
diagramme, 5-1

espacement régulier. *Voir l'Aide
LabVIEW.*
étiquetage, 4-18
création. *Voir l'Aide LabVIEW.*
édition. *Voir l'Aide LabVIEW.*
redimensionnement. *Voir l'Aide
LabVIEW.*
face-avant et terminaux du
diagramme, 5-1
groupement et verrouillage sur la
face-avant, 4-5
insertion dans des palettes. *Voir l'Aide
LabVIEW.*
insertion sur la face-avant à l'aide
d'ActiveX, 18-4
insertion sur le diagramme. *Voir l'Aide
LabVIEW.*
masquage de la face-avant
Voir aussi l'Aide LabVIEW.
éléments optionnels, 4-2
masqués sur la face-avant. *Voir l'Aide
LabVIEW.*
mise à l'échelle sur la face-avant, 4-6
paramétrage de l'ordre de navigation sur
la face-avant, 4-4
recherche. *Voir l'Aide LabVIEW.*
redimensionnement sur la face-avant, 4-6
par rapport à la taille de la fenêtre, 4-6
remplacement sur la face-avant, 4-2
remplacement sur le diagramme. *Voir
l'Aide LabVIEW.*
réorganisation. *Voir l'Aide LabVIEW.*
sélection. *Voir l'Aide LabVIEW.*
sous-titres sur la face-avant, 4-19
création. *Voir l'Aide LabVIEW.*
transparents. *Voir l'Aide LabVIEW.*
objets masqués de la face-avant. *Voir l'Aide
LabVIEW.*
OLE pour le protocole DataSocket de contrôle
de procédé, 17-3
opérateurs. *Reportez-vous à nœuds.*

- options
 - enregistrement, 3-7
 - paramétrage, 3-6
 - Voir aussi l'Aide LabVIEW.*
 - options de l'environnement de travail
 - enregistrement, 3-7
 - paramétrage, 3-6
 - Voir aussi l'Aide LabVIEW.*
 - ordre d'exécution
 - contrôle avec des structures
 - Séquence, 8-10
 - ordre de la face-avant
 - paramétrage, 4-4
 - ordre de navigation
 - paramétrage, 4-4
 - ordre des éléments de cluster, 9-16
 - modification, 9-16
 - Voir aussi l'Aide LabVIEW.*
 - oscillographe, 11-7
 - Outil de publication sur le Web, 17-10
 - outil Point d'arrêt
 - mise au point des VIs, 6-6
 - mise en surbrillance des points d'arrêt.
 - Voir l'Aide LabVIEW.*
 - outil Sonde
 - mise au point des VIs, 6-5
 - outils
 - palette, 3-2
 - sélection manuelle. *Voir l'Aide LabVIEW.*
 - ouverture des VIs en mode exécution. *Voir l'Aide LabVIEW.*
- P**
- palettes, 3-1
 - Commandes, 3-1
 - personnalisation, 3-4
 - Fonctions, 3-1
 - personnalisation, 3-4
 - insertion d'objets. *Voir l'Aide LabVIEW.*
 - mise à jour. *Voir l'Aide LabVIEW.*
 - modifications. *Voir l'Aide LabVIEW.*
 - navigation et recherche, 3-2
 - organisation, 3-5
 - Outils, 3-2
 - partage. *Voir l'Aide LabVIEW.*
 - personnalisation, 3-4
 - référence. *Voir l'Aide LabVIEW.*
 - sélecteur de couleurs, 4-10
 - vues, 3-6
 - paramétrage
 - niveau de coopération. *Voir l'Aide LabVIEW.*
 - options de l'environnement de travail, 3-6
 - Voir aussi l'Aide LabVIEW.*
 - paramètre de chaîne de format, 9-4
 - paramètres
 - types de données (table), 5-2
 - paramètres dupliqués, 13-14
 - partage
 - des fichiers, 7-2
 - données en direct avec d'autres VIs et applications, 17-2
 - données en direct par programmation, 17-7
 - VIs, 7-13
 - vues de palettes. *Voir l'Aide LabVIEW.*
 - partage de fichiers, 7-2
 - pas à pas dans les VIs
 - mise au point des VIs, 6-5
 - Pas un chemin, 4-12
 - performance
 - désactivation des outils de mise au point, 6-8
 - options, 3-6
 - variables globales et locales, 10-4
 - personnalisation
 - apparence et comportement des VIs, 15-1
 - codes d'erreur. *Voir l'Aide LabVIEW.*
 - environnement de travail, 3-4
 - menus, 15-2
 - objets de la face-avant, 4-1

- palettes, 3-4
- pires
 - données variant, 5-19
- planification des projets, 7-1
- planification du projet, 7-1
- plusieurs threads
 - exécution. *Voir l'Aide LabVIEW.*
- points
 - coercition, 5-14
- points de coercition, 5-14
- police de l'application, 4-20
- police du système, 4-20
- polices
 - application, 4-20
 - dialogue, 4-20
 - options, 3-7
 - paramètres, 4-19
 - système, 4-20
- polymorphes
 - commandes et indicateurs
 - type de données (table), 5-4
 - fonctions, B-1
 - nœuds d'expression, 20-5
 - unités, B-1
 - VI, 5-15
 - construction, 5-16
 - édition des menus locaux. *Voir l'Aide LabVIEW.*
 - sélection d'une instance par défaut. *Voir l'Aide LabVIEW.*
 - suppression de sous-VI. *Voir l'Aide LabVIEW.*
- PPC Toolbox, 17-16
- préférences. *Reportez-vous à options.*
- premier entré, premier sorti
 - données variant, 5-19
- propriétés
 - ActiveX, 18-1
 - paramétrage, 18-5
 - par programmation, 18-5
 - visualisation, 18-5

- protection par mot de passe, 7-13
- protocole DataSocket dstp, 17-3
- protocole DataSocket fichier, 17-4
- protocole DataSocket ftp, 17-4
- protocole DataSocket logos, 17-4
- protocole DataSocket opc, 17-3
- protocoles
 - communication de bas niveau, 17-15
 - DataSocket, 17-3
- publication de VIs sur le Web, 17-9

R

- raccourcis clavier, 4-3
 - contrôler les boutons, 4-3
 - paramétrage de l'ordre de navigation, 4-4
- rafraîchissements progressifs
 - pendant le traçage. *Voir l'Aide LabVIEW.*
 - pour les graphes, 11-6
- rapports
 - génération
 - clusters d'erreur. *Voir l'Aide LabVIEW.*
 - impression, 14-6
 - VI de génération de rapport, 14-6
- recherche
 - commandes, VI et fonctions sur les palettes, 3-2
 - erreurs, 6-2
 - hiérarchie du VI. *Voir l'Aide LabVIEW.*
 - objets, texte et VI. *Voir l'Aide LabVIEW.*
 - versions PDF de la documentation labVIEW, 1-1
- récupération des données
 - par programmation, 13-20
 - utilisation de sous-VI, 13-20
 - utilisation des fonctions d'E/S sur fichier, 13-22
- redimensionnement
 - clusters. *Voir l'Aide LabVIEW.*
 - constantes définies par l'utilisateur, 5-5

étiquettes. *Voir l'Aide LabVIEW.*
fonctions. *Voir l'Aide LabVIEW.*
nœuds. *Voir l'Aide LabVIEW.*
objets de la face-avant, 4-6
 par rapport à la taille de la fenêtre, 4-6
tableaux. *Voir l'Aide LabVIEW.*
tables. *Voir l'Aide LabVIEW.*
références des commandes, 16-9
 création. *Voir l'Aide LabVIEW.*
 de type strict, 16-9
 partiellement spécifiés, 16-10
refnums
 automation, 18-2
 commande, 16-9
 commandes et indicateurs, 4-17
 type de données (table), 5-3
 utilisation. *Voir l'Aide LabVIEW.*
 de type strict, 16-8
 E/S sur fichier, 13-1
 nœud d'appel par référence, 16-8
refnums de commande partiellement
 spécifiés, 16-10
refnums de type strict
 commande, 16-9
 VI, 16-8
registres à décalage, 8-6
 données par défaut, 6-9
remplacement
 des objets de la face-avant, 4-2
 éléments dans des tableaux. *Voir l'Aide
 LabVIEW.*
 objets sur le diagramme. *Voir l'Aide
 LabVIEW.*
 texte dans des chaînes. *Voir l'Aide
 LabVIEW.*
réorganisation d'objets. *Voir l'Aide LabVIEW.*
réparation
 VIs, 6-2
 techniques de mise au point, 6-4

répertoire de structure et support
 menus, A-2
 project, A-2
 resource, A-2
 templates, A-2
 WWW, A-2
répertoire exercices et exemples
 activity, A-2
 examples, A-2
 tutorial, A-2
répertoires
 conversion des bibliothèques en. *Voir
 l'Aide LabVIEW.*
 conversion en bibliothèques. *Voir l'Aide
 LabVIEW.*
répétition
 blocs of code
 boucles For, 8-2
 boucles While, 8-3
représentation des couleurs, 11-13
réservoirs
 Voir aussi numériques.
 commandes et indicateurs à glissières, 4-9
résolutions écran, 4-22
ressources techniques, E-1
revenir aux dernières versions enregistrées.
 Voir l'Aide LabVIEW.
RTF
 enregistrement dans un fichier, 14-3

S
saisie semi-automatique, 4-14
 commandes de type menu déroulant, 4-15
 listes déroulantes, 4-14
schéma pour le format XML, 9-8
sélection
 fils de liaison, 5-14
 instance par défaut d'un VI polymorphe.
 Voir l'Aide LabVIEW.
 objets. *Voir l'Aide LabVIEW.*

- outils, de façon manuelle. *Voir l'Aide LabVIEW.*
- Séparateur décimal
 - localisé. *Voir l'Aide LabVIEW.*
- Séparateur décimal localisé. *Voir l'Aide LabVIEW.*
- Serveur Web, 17-9
 - activation, 17-9
 - clients pour les faces-avant distantes
 - LabVIEW, 17-12
 - multiples, 17-11
 - navigateur Web, 17-13
 - contrôle de VIs, 17-11
 - licences pour servir des faces-avant distantes, 17-12
 - options, 17-9
 - visualisation de faces-avant, 17-11
- serveurs
 - ActiveX, 18-6
 - Ccnfiguration pour les faces-avant distantes, 17-11
 - LabVIEW, 17-13
 - navigateur Web, 17-13
- situations de compétition, 10-5
- sous-palettes
 - construction d' ActiveX, 3-6
 - création. *Voir l'Aide LabVIEW.*
 - déplacement. *Voir l'Aide LabVIEW.*
 - organisation, 3-5
- sous-programme. *Reportez-vous au sous-VIs.*
- sous-titres, 4-19
 - création. *Voir l'Aide LabVIEW.*
 - info-bulles des sous-VIs. *Voir l'Aide LabVIEW.*
- sous-VIs, 7-4
 - affichage de la chaîne d'appelants, 6-7
 - affichage des noms lorsque placés. *Voir l'Aide LabVIEW.*
 - conception, 7-9
 - construction, 7-4
 - contrôle du comportement, 7-3
 - copie, A-3
 - création, 7-9
 - situations à éviter. *Voir l'Aide LabVIEW.*
 - détermination de l'instance actuelle, 6-7
 - face-avant, 7-9
 - faces-avant distantes, 17-14
 - hiérarchie, 7-10
 - impression de VIs, 14-8
 - récupération des données de la face-avant, 13-20
 - sous-titres des commandes pour les info-bulles. *Voir l'Aide LabVIEW.*
 - suppression d'un sous-VI d'un VI polymorphe. *Voir l'Aide LabVIEW.*
 - suspension de l'exécution, 6-6
 - VIs polymorphes, 5-15
- structure des répertoires de LabVIEW, A-1
 - Macintosh, A-3
 - organisation, A-1
- structures, 8-1
 - boucles For, 8-2
 - boucles While, 8-3
 - câblage *Voir l'Aide LabVIEW.*
 - Condition, 8-7
 - effacement. *Voir l'Aide LabVIEW.*
 - Événement, 8-14
 - mise au point. *Voir l'Aide LabVIEW.*
 - Séquence, 8-10
 - suppression. *Voir l'Aide LabVIEW.*
 - sur le diagramme, 2-4
 - utilisation. *Voir l'Aide LabVIEW.*
 - variables globales, 10-2
 - variables locales, 10-1
- structures Condition, 8-7
 - gestion d'erreur, 6-12
 - spécification d'une condition par défaut, 8-8
 - terminaux de sélecteurs
 - valeurs, 8-8
 - terminaux de sélection, 8-8

- types de données, 8-8
- utilisation. *Voir l'Aide LabVIEW.*
- Structures Événement
 - utilisation. *Voir l'Aide LabVIEW.*
- Structures Événement, 8-14
- structures Séquence, 8-10
 - Accès aux valeurs avec des variables globales et locales, 10-4
 - contrôle de l'ordre d'exécution, 5-23
 - usage excessif, 8-11
 - utilisation. *Voir l'Aide LabVIEW.*
- suiti du développement. *Reportez-vous à documentation des VIs.*
- support international, E-2
- support sur le Web, E-1
- suppression
 - fils de liaison brisés, 5-14
 - objets sur la face-avant ou le diagramme. *Voir l'Aide LabVIEW.*
 - sous-VIs de VIs polymorphes. *Voir l'Aide LabVIEW.*
 - structures. *Voir l'Aide LabVIEW.*
 - terminaux provenant des fonctions, 5-11
 - VIs des bibliothèques. *Voir l'Aide LabVIEW.*
- suspension de l'exécution
 - mise au point des VIs, 6-6
- synchronisation
 - contrôle, 8-7
- T**
- tableaux, 9-9
 - auto-indexation des boucles, 8-4
 - commandes et indicateurs, 4-12
 - type de données (table), 5-3
 - comparaison, C-2
 - constantes, 9-12
 - conversion des clusters. *Voir l'Aide LabVIEW.*
 - création, 9-12
 - déplacement. *Voir l'Aide LabVIEW.*
 - dimensions, 9-9
 - données par défaut, 6-9
 - exemples, 9-9
 - tableaux 1D, 9-9
 - tableaux 2D, 9-11
 - fonctions, 5-8
 - indices, 9-9
 - affichage, 9-13
 - insertion d'éléments. *Voir l'Aide LabVIEW.*
 - polymorphisme, B-5
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - remplacement des éléments. *Voir l'Aide LabVIEW.*
 - restrictions, 9-12
 - suppression d'éléments. *Voir l'Aide LabVIEW.*
 - taille des, 6-9
 - variables globales, 10-6
- tables, 9-3
 - utilisation. *Voir l'Aide LabVIEW.*
- tables de pixels, 12-5
- taille de la fenêtre
 - définition. *Voir l'Aide LabVIEW.*
- TCP, 17-15
 - VI Serveur, 16-1
- terminaux, 2-3
 - affichage, 5-2
 - affichage des info-bulles. *Voir l'Aide LabVIEW.*
 - ajout de fonctions, 5-11
 - câblage, 5-11
 - commandes et indicateurs (table), 5-2
 - conditionnels, 8-3
 - constantes, 5-4
 - décompte, 8-2
 - auto-indexation pour définir, 8-5
 - diagramme, 5-2

- itération
 - boucles For, 8-2
 - boucles While, 8-3
 - motifs, 7-7
 - objets de la face-avant et, 5-1
 - points de coercition, 5-14
 - recherche. *Voir l'Aide LabVIEW.*
 - sélecteur, 8-8
 - suppressions en provenance de fonctions, 5-11
 - variable locale de séquence, 8-11
 - terminaux conditionnels, 8-3
 - terminaux d'itération
 - boucles For, 8-2
 - boucles While, 8-3
 - terminaux de décompte, 8-2
 - auto-indexation pour définir, 8-5
 - terminaux de sélecteurs
 - valeurs, 8-8
 - terminaux de sélection, 8-8
 - terminaux de variable locale de séquence, 8-11
 - terminaux récepteurs. *Reportez-vous à indicateurs.*
 - terminaux source. *Reportez-vous à commandes.*
 - texte
 - commandes de type menu déroulant, 4-15
 - faire glisser et déposer. *Voir l'Aide LabVIEW.*
 - formatage, 4-19
 - recherche. *Voir l'Aide LabVIEW.*
 - zones de saisie, 4-11
 - thermomètres
 - Voir aussi* numériques.
 - commandes et indicateurs à glissières, 4-9
 - threads
 - exécution de plusieurs. *Voir l'Aide LabVIEW.*
 - titres des fenêtres dans la palette de Fonctions. *Voir l'Aide LabVIEW.*
 - toolsets, 1-1
 - dans les palettes, 3-6
 - toolsets additionnels, 1-1
 - dans les palettes, 3-6
 - touches de fonction par défaut
 - ignorer par défaut. *Voir l'Aide LabVIEW.*
 - traçage
 - Reportez-vous à* graphiques.
 - rafraîchissements progressifs. *Voir l'Aide LabVIEW.*
 - tracé de lignes
 - lissage, 11-2
 - transfert des VIs., 7-13
 - Transmission Control Protocol, 17-15
 - transparent
 - étiquettes. *Voir l'Aide LabVIEW.*
 - objets. *Voir l'Aide LabVIEW.*
 - tunnels, 8-1
 - entrée et sortie, 8-9
 - tutorial, 1-2
 - types de données
 - commandes et indicateurs (table), 5-2
 - conversion à partir d'XML, 9-8
 - conversion en XML, 9-7
 - données waveform, 11-19
 - HiQ (table), 20-6
 - impression, 14-3
 - MATLAB (table), 20-6
 - structures Condition, 8-8
- ## U
- UDP, 17-15
 - unités de mesures, 5-19
 - URL pour DataSocket, 17-3
 - User Datagram Protocol, 17-15
 - utilisation de la touche tab pour naviguer entre les objets de la face-avant, 4-4
 - Utilitaire de configuration NI-DAQ, 1-4

V

variables

- globales, 10-2
 - création, 10-3
 - initialisation, 10-5
 - lecture et écriture, 10-3
 - mémoire, 10-6
 - situations de compétition, 10-5
 - utilisation mesurée, 10-4

locales, 10-1

- création, 10-2
- initialisation, 10-5
- lecture et écriture, 10-3
- mémoire, 10-6
- situations de compétition, 10-5
- utilisation mesurée, 10-4

variables globales, 10-2

- création, 10-3
- initialisation, 10-5
- lecture et écriture, 10-3
- mémoire, 10-6
- situations de compétition, 10-5
- utilisation mesurée, 10-4

variables globales d'écriture, 10-3

variables globales de lecture, 10-4

variables locales, 10-1

- création, 10-2
- initialisation, 10-5
- lecture et écriture, 10-3
- mémoire, 10-6
- recherche d'objets ou de terminaux. *Voir l'Aide LabVIEW.*
- situations de compétition, 10-5
- utilisation mesurée, 10-4

variables locales d'écriture, 10-3

variables locales de lecture, 10-4

variables statiques. *Reportez-vous à registres à décalage.*

vérification de l'espace disque disponible. *Voir l'Aide LabVIEW.*

vérification de type stricte, 5-20

verrouillage

- objets de la face-avant, 4-5
- VIs. *Voir l'Aide LabVIEW.*

versions

- comparaison, 7-2
- enregistrement des VIs pour la précédente version, 7-12
- revenir aux dernières enregistrées. *Voir l'Aide LabVIEW.*

versions antérieures

- enregistrement des VIs, 7-12

VI Serveur, 16-1

- appel à distance des VIs, 16-1
- appel d'autres instances de LabVIEW sur le Web, 16-1
- applications à distance, 16-9
- caractéristiques, 16-1
- construire des applications, 16-2
- contrôle des objets de la face-avant, 16-9
- manipulation des paramètres de VI, 16-4
- mise en réseau et, 17-1
- nœud d'appel par référence, 16-8
- nœud de méthode, 16-5
- nœud de propriété, 16-4
- Objet d'application, 16-3
- Objet du VI, 16-3
- refnums de VI de type strict, 16-8

VI System Exec, 17-17

VIs, 2-1

- ajouter à des bibliothèques, 3-5
- appel à distance, 16-1
- appeler de façon dynamique, 16-7
- appellation, 7-12
- bibliothèques, 7-11
- brisés, 6-2
- charger de façon dynamique, 16-7
- comparer des versions, 7-2
- configuration de l'apparence et du comportement, 15-1
- construction, 7-1

- contrôle lorsqu'ils sont appelés comme sous-VIs, 7-3
 - contrôle par programmation, 16-1
 - contrôle sur le Web, 17-9
 - copie, A-3
 - correction, 6-2
 - création d'info-bulles, 14-2
 - création de descriptions, 14-2
 - développement, 7-1
 - déverrouillage. *Voir l'Aide LabVIEW.*
 - distribution, 7-13
 - documentation, 14-1
 - enregistrement, 7-10
 - exécutables
 - mise au point. *Voir l'Aide LabVIEW.*
 - exécution, 6-1
 - exemples, 1-4
 - faire glisser et déposer. *Voir l'Aide LabVIEW.*
 - gestion d'erreur, 6-10
 - hiérarchie, 7-10
 - impression, 14-5
 - lancement de la ligne de commande. *Voir l'Aide LabVIEW.*
 - localisation., 7-13
 - marquage comme étant de niveau principal dans des bibliothèques. *Voir l'Aide LabVIEW.*
 - mise à jour, 7-12
 - ouverture en mode exécution. *Voir l'Aide LabVIEW.*
 - partage, 7-13
 - polymorphes, 5-15
 - publication sur le Web, 17-9
 - recherche. *Voir l'Aide LabVIEW.*
 - référence. *Voir l'Aide LabVIEW.*
 - refnums de type strict, 16-8
 - retrait des bibliothèques. *Voir l'Aide LabVIEW.*
 - revenir aux dernières versions enregistrées. *Voir l'Aide LabVIEW.*
 - techniques de mise au point, 6-4
 - transfert, 7-13
 - verrouillage. *Voir l'Aide LabVIEW.*
 - VIs brisés
 - affichage des erreurs, 6-2
 - causes courantes, 6-3
 - correction, 6-2
 - VIs de fichiers de configuration
 - format, 13-15
 - lecture et écriture de fichiers .ini, 13-14
 - spécial, 13-15
 - VIs exécutables
 - mise au point. *Voir l'Aide LabVIEW.*
 - VISA
 - transmission de noms de ressource, 4-16
 - visualisation
 - Reportez-vous aussi à affichage.*
 - vitesse d'exécution
 - contrôle, 8-7
 - vitesse de clignotement. *Voir l'Aide LabVIEW.*
 - voyants sur la face-avant, 4-11
 - vues, 3-6
 - création, 3-5
 - édition, 3-5
 - effacement. *Voir l'Aide LabVIEW.*
 - modifications. *Voir l'Aide LabVIEW.*
 - partage. *Voir l'Aide LabVIEW.*
 - VXI
 - configuration, 1-4
 - VIs, 1-3
- ## W
- waveforms
 - écrire dans des fichiers, 13-12
 - lecture de fichiers, 13-13
 - Web
 - appel d'autres instances de LabVIEW, 16-1
 - contrôle de VIs, 17-11

création de documents HTML, 17-10
publication de VIs, 17-9
visualisation de faces-avant, 17-11

X

XML

conversion à partir de, 9-8
Conversion de types de données en, 9-7
exemple, 9-6
schéma, 9-8

Z

zoom sur des graphes et graphes déroulants.
Voir l'Aide LabVIEW.