

LabVIEW™

Manuel de l'utilisateur

Filiales francophones

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

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. : 33 0 1 48 14 24 24	Fax : 33 0 1 48 14 24 14	Tél. : 33 0 5 61 31 62 80 (Toulouse)
Suisse	Tél. : 41 56 200 51 51	Fax : 41 56 200 51 55	Tél. : 41 21 320 51 51 (Lausanne)
Belgique	Tél. : 32 0 2 757 00 20	Fax : 32 0 2 757 03 11	Tél. : 405 020 (Luxembourg)
Canada (Québec)	Tél. : 514 288 5722	Fax : 514 694 4399	

Les filiales

Afrique du Sud 27 0 11 805 8197, Allemagne 49 0 89 741 31 30, Australie 1800 300 800,
Autriche 43 0 662 45 79 90 0, Brésil 55 11 3262 3599, Canada (Calgary) 403 274 9391,
Canada (Montreal) 514 288 5722, Canada (Ottawa) 613 233 5949, Canada (Toronto) 905 785 0085,
Canada (Vancouver) 514 685 7530, Chine 86 021 6555 7838, Corée 82 02 3451 3400,
Danemark 45 45 76 26 00, Espagne 34 91 640 0085, Finlande 385 0 9 725 725 11,
Grande-Bretagne 44 0 1635 523545, Grèce 30 2 10 42 96 427, Inde 91 80 51190000, Israël 972 0 3 6393737,
Italie 39 02 413091, Japon 81 3 5472 2970, Malaisie 603 9131 0918, Mexique 001 800 010 0793,
Norvège 47 0 66 90 76 60, Nouvelle-Zélande 1800 300 800, Pays-Bas 31 0 348 433 466,
Pologne 48 0 22 3390 150, Portugal 351 210 311 210, République tchèque 420 2 2423 5774,
Russie 7 095 238 7139, Singapour 65 6226 5886, Slovaquie 386 3 425 4200, Suède 46 0 8 587 895 00,
Taiwan 886 2 2528 7227, Thaïlande 662 992 7519

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 *Support technique et services professionnels*. 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 contre toute panne d'exécution des instructions de programmation du fait d'un défaut du matériel ou d'un défaut de fabrication, et ce pendant une durée de 90 jours à compter de leur date d'expédition, cette date pouvant être prouvée au moyen de reçus ou autres justificatifs. Les supports n'exécutant pas les instructions de programmation seront, au choix de National Instruments, soit réparés, soit remplacés, sous réserve que National Instruments reçoive notification de ces défauts pendant la période de garantie mentionnée ci-dessus. National Instruments ne garantit pas que le fonctionnement du logiciel sera ininterrompu ou exempt d'erreurs.

Les produits renvoyés à National Instruments dans le cadre de la garantie ci-dessus ne seront acceptés que si un numéro d'Autorisation de Retour de Matériel (RMA) a été au préalable obtenu auprès de National Instruments et que ce numéro est clairement apposé à l'extérieur de l'emballage dans lequel les produits sont retournés. National Instruments prendra à sa charge les frais de renvoi au propriétaire des pièces couvertes par la présente garantie.

National Instruments considère que les informations contenues dans le présent document sont exactes. Ce document a été soigneusement revu afin d'en vérifier l'exactitude technique. Si des erreurs techniques ou typographiques figurent dans ce document, National Instruments se réserve le droit d'apporter les modifications nécessaires aux éditions ultérieures de ce document sans être tenu d'en informer au préalable les personnes qui possèdent la présente édition. S'il suspecte des erreurs, le lecteur doit en aviser National Instruments. National Instruments ne sera en aucun cas responsable des dommages causés par ou en relation avec ce document ou les informations qu'il contient.

En dehors de ce qui est expressément indiqué ci-dessus, National Instruments ne donne aucune garantie, explicite ou implicite, et exclut spécifiquement toute garantie quant à la qualité marchande ou l'aptitude à une utilisation particulière. La réparation pouvant être obtenue par le client pour les dommages résultant d'une faute ou de la négligence de National Instruments sera limitée au prix d'achat du produit en cause. National Instruments ne sera pas responsable des dommages résultant de la perte de données, de la perte de profits, de l'utilisation de produits ou des dommages accessoires ou indirects, même si National Instruments a été averti de l'éventualité de tels dommages. Cette limitation de responsabilité sera applicable quelle que soit l'action engagée, que son fondement soit contractuel ou quasi-délictuel, y compris si le fondement invoqué est la négligence. Aucune réclamation ne sera acceptée plus d'un an après la survenance du fondement de cette réclamation. National Instruments ne sera en aucun cas responsable des retards d'exécution résultant d'événements indépendants de sa volonté. La présente garantie ne couvre pas les dommages, défauts, dysfonctionnements ou défaillances de service résultant du non-respect par le propriétaire des instructions fournies par National Instruments concernant l'installation, l'utilisation ou la maintenance du produit; de la modification du produit par le propriétaire; de l'abus, la mauvaise utilisation ou des actes de négligence de la part du propriétaire; ou des pannes d'électricité ou des surtensions, d'incendies, d'inondations, d'accidents, d'actes de tiers ou d'autres événements indépendants de la volonté de National Instruments.

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

CVI™, DataSocket™, DIAdem™, IMAQ™, IVI™, LabVIEW™, Measurement Studio™, National Instruments™, NI™, ni.com™, NI-DAQ™, NI Developer Zone™, NI-VXI™ 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 **Aides/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 modèles, VI s d'exemple et outils LabVIEW	1-4
VI s modèles LabVIEW	1-4
VI s d'exemple LabVIEW	1-5
Outils LabVIEW	1-5

Chapitre 2

Introduction aux instruments virtuels

Face-avant	2-1
Diagramme.....	2-2
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	3-2
Parcourir les palettes de Commandes et de Fonctions.....	3-2
PaLETTE d'outils	3-3
Menus et barre d'outils	3-4
Menus	3-4
Menus locaux	3-4
Menus locaux en mode Exécution.....	3-4

Barre d'outils.....	3-5
Aide contextuelle.....	3-5
Personnalisation de votre environnement de travail.....	3-6
Personnalisation des palettes de Commandes et de Fonctions	3-6
Ajout de VIs et de commandes aux sous-palettes Utilisateur et Drivers d'instruments.....	3-7
Création et édition d'un type de palette personnalisé.....	3-7
Comment LabVIEW enregistre les types.....	3-8
Construction des sous-palettes ActiveX.....	3-8
Représentation des toolsets et des modules dans les palettes	3-8
Paramétrage des options de l'environnement de travail	3-9
Comment LabVIEW enregistre les options.....	3-9
Windows.....	3-9
Mac OS.....	3-9
UNIX.....	3-10

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 de commandes en indicateurs et inversement.....	4-2
Remplacement des objets de la face-avant	4-3
Configuration de la face-avant	4-3
Paramétrage des raccourcis clavier des commandes.....	4-4
Contrôler le comportement d'un bouton avec le raccourci clavier.....	4-4
Définition de l'ordre de défilement par tabulation des objets de la face-avant	4-5
Coloriage des objets	4-5
Utilisation d'images importées	4-6
Alignement et répartition d'objets	4-6
Groupement et verrouillage des objets.....	4-7
Redimensionnement des objets.....	4-7
Mise à l'échelle des objets de la face-avant.....	4-8
Ajout d'espace dans la face-avant sans redimensionner la fenêtre	4-10
Commandes et indicateurs de la face-avant	4-10
Commandes et indicateurs classiques et 3D	4-10
Glissières, boutons rotatifs, cadrans, afficheurs numériques et horodatage ...	4-11
Commandes et indicateurs à glissières	4-11
Commandes et indicateurs rotatifs	4-12
Commandes et indicateurs numériques	4-12
Formatage numérique.....	4-12
Commande et indicateurs d'horodatage	4-13

Boîtes de couleur.....	4-13
Rampes de couleurs	4-13
Graphes et graphes déroulants.....	4-14
Boutons, commutateurs et voyants.....	4-14
Affichages des zones de saisie de texte, étiquettes et chemins	4-14
Commandes et indicateurs de type chaîne	4-14
Commandes de type chaîne déroulante.....	4-15
Commandes et indicateurs de type chemin.....	4-16
Chemins invalides.....	4-16
Chemins vides.....	4-16
Commandes et indicateurs de type tableau et cluster.....	4-17
Commandes de type liste déroulante, de type arbre et tables.....	4-17
Listes déroulantes.....	4-17
Commandes arbre	4-18
Tables.....	4-19
Commandes et indicateurs de type énumération et menu déroulant	4-19
Commandes de menu déroulant.....	4-19
Commandes de type énumération	4-20
Propriétés avancées des commandes et indicateurs de type énumération	4-21
Commandes de type Conteneur.....	4-21
Commandes onglet.....	4-22
Commandes de face-avant secondaire	4-22
Commandes et indicateurs de nom d'E/S.....	4-23
Commande waveform	4-24
Commande de waveform numérique	4-24
Commande de données numériques.....	4-24
Conversion des données en données numériques	4-26
Acquisition d'un sous-ensemble numérique.....	4-27
Ajout d'échantillons et de signaux numériques.....	4-28
Compression de données numériques.....	4-28
Recherche d'un pattern	4-28
Références à des objets ou à des applications	4-29
Commandes et indicateurs de boîte de dialogue	4-29
Étiquetage	4-30
Sous-titres	4-31
Caractéristiques du texte.....	4-31
Conception d'interfaces utilisateur	4-33
Utilisation des commandes et indicateurs de la face-avant	4-33
Conception des boîtes de dialogue	4-34
Sélection de la taille de l'écran.....	4-35

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	5-5
Constantes universelles	5-6
Les constantes définies par l'utilisateur	5-6
Nœuds du diagramme	5-7
Sommaire des fonctions	5-8
Fonctions numériques	5-8
Fonctions booléennes	5-9
Fonctions chaîne	5-9
Fonctions tableau	5-9
Fonctions cluster	5-10
Fonctions de comparaison	5-10
Fonctions de temps et dialogue	5-10
Fonctions d'E/S sur fichiers	5-10
Fonctions de waveform	5-11
Fonctions de contrôle d'applications	5-11
Fonctions avancées	5-11
Ajout de terminaux aux fonctions	5-12
Utilisation des fils de liaison pour relier les objets du diagramme	5-12
Câblage automatique des objets	5-14
Câblage manuel des objets	5-15
Routage des fils de liaison	5-15
Sélection des fils de liaison	5-16
Correction des fils de liaison brisés	5-16
Points de coercion	5-17
VI et fonctions polymorphes	5-18
VIs polymorphes	5-18
Construction de VIs polymorphes	5-19
Fonctions polymorphes	5-21
VI Express	5-22
Création de VI Express en sous-VIs	5-22
Type de données dynamiques	5-22
Convertir des données dynamiques	5-24
Conversion en données dynamiques	5-25
Manipulation de données variant	5-25
Unités numériques et vérification de type stricte	5-27
Unités et vérification de type stricte	5-27

Flux des données dans le diagramme	5-29
Dépendance des données et dépendance artificielle des données	5-30
Dépendance de données manquante	5-31
Flux des données et gestion de mémoire	5-32
Conception du diagramme	5-32

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 de VIs brisés	6-2
Causes fréquentes de VIs brisés	6-3
Techniques de mise au point	6-4
Mode Animation	6-6
Mode pas à pas	6-6
Outil Sonde	6-7
Types de sondes	6-7
Générique	6-7
Utilisation d'indicateurs pour afficher les données	6-8
Sondes fournies	6-8
Sondes personnalisées	6-9
Points d'arrêt	6-9
Suspension de l'exécution	6-10
Détermination de l'instance actuelle d'un sous-VI	6-11
Mise "en commentaire" de sections particulières du diagramme	6-11
Désactivation des outils de mise au point	6-12
Données indéfinies ou inattendues	6-12
Données par défaut dans des boucles	6-13
Boucles For	6-13
Données par défaut dans les tableaux	6-13
Éviter des données indéfinies	6-14
Détection et gestion d'erreur	6-14
Détection d'erreurs	6-15
Gestion d'erreur	6-15
Clusters d'erreur	6-16
Utilisation des boucles While pour la gestion d'erreur	6-16
Utilisation des structures Condition pour la gestion d'erreur	6-17

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
VIs modèles	7-2
Création de VIs modèles	7-3
Autres types de document.....	7-3
Utilisation des VIs et fonctions intégrés.....	7-3
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.....	7-4
Construction de VIs qui communiquent avec d'autres applications.....	7-4
Sous-VIs	7-5
Recherche d'opérations courantes	7-6
Configuration du connecteur.....	7-7
Définition des entrées et sorties nécessaires, recommandées et facultatives	7-8
Création d'une icône	7-9
Affichage de sous-VIs et de VIs Express sous forme d'icônes ou de nœuds extensibles.....	7-10
Création de sous-VIs à partir de portions d'un VI.....	7-11
Conception de sous-VIs	7-11
Affichage de la hiérarchie des VIs	7-12
Enregistrement des VIs.....	7-13
Avantages de l'enregistrement des VIs en tant que fichiers individuels	7-13
Avantages de l'enregistrement des VIs dans des bibliothèques	7-13
Gestion des VIs dans les bibliothèques.....	7-14
Appellation des VIs.....	7-15
Enregistrement au format antérieur.....	7-15
Distribution des VIs.....	7-15
Construction d'applications autonomes et de bibliothèques partagées	7-16

PARTIE II

Construction et édition de VIs

Chapitre 8

Boucles et structures

Structures boucles For et boucles 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
Utilisation de boucles pour construire des tableaux	8-6
Registres à décalage et Nœud de rétroaction dans les boucles.....	8-6
Registres à décalage	8-7
Registres à décalage empilés.....	8-8
Remplacement des registres à décalage par des tunnels	8-9
Remplacement des tunnels par des registres à décalage	8-9
Nœud de rétroaction	8-10
Initialisation des nœuds de rétroaction.....	8-11
Remplacement de registres à décalage par un nœud de rétroaction.....	8-12
Contrôle de la vitesse.....	8-12
Structures Condition et Séquence	8-12
Structures Condition	8-13
Valeurs du sélecteur de condition et types de données	8-13
Tunnels d'entrée et de sortie	8-15
Utilisation des structures Condition pour la gestion d'erreur	8-15
Structures Séquence.....	8-15
Structure Séquence déroulée.....	8-16
Structure Séquence empilée	8-16
Utilisation des structures Séquence.....	8-17
Éviter un usage excessif des structures Séquence.....	8-17
Remplacement de structures Séquence	8-19

Chapitre 9

Programmation événementielle

Que sont les événements ?	9-1
Pour quelle raison utiliser les événements ?	9-2
Composantes de la structure Événement	9-3
Événements de type filtre et notification	9-5
Utilisation des événements dans LabVIEW	9-6
Enregistrement statique d'événements.....	9-9
Enregistrement dynamique d'événements	9-10
Exemple d'événement dynamique.....	9-13
Modification dynamique de l'enregistrement	9-13
Événements utilisateur.....	9-15
Création et enregistrement d'événements utilisateur	9-15
Génération d'événements utilisateur.....	9-16
Désenregistrement d'événements utilisateur	9-17
Exemple d'événement utilisateur.....	9-17

Chapitre 10

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

Chaînes	10-1
Chaînes sur la face-avant	10-2
Types d'affichage de chaîne	10-2
Tables	10-3
Édition de chaînes par programmation	10-3
Formatage de chaînes	10-4
Spécificateurs de format	10-4
Numériques et chaînes	10-5
Conversion de données en données XML et vice versa.....	10-6
Utilisation de types de données basés sur le format XML	10-7
Schéma XML de LabVIEW	10-8
Groupement des données avec des tableaux et des clusters	10-9
Tableaux	10-9
Indices	10-9
Exemples de tableaux	10-9
Restrictions pour les tableaux	10-12
Création de commandes, d'indicateurs et de constantes tableau	10-12
Affichage de l'indice du tableau	10-13
Fonctions de tableaux	10-14
Redimensionnement automatique des fonctions de tableaux	10-14
Clusters	10-15

Chapitre 11

Variables globales et locales

Variables locales	11-1
Création de variables locales	11-2
Variables globales	11-2
Création de variables globales	11-3
Variables de lecture et d'écriture	11-4
Utilisation mesurée des variables globales et locales	11-4
Initialisation des variables globales et locales	11-5
Situations de compétition	11-5
Considérations sur la mémoire lors de l'utilisation de variables locales	11-6
Considérations sur la mémoire lors de l'utilisation de variables globales	11-6

Chapitre 12

Graphes et graphes déroulants

Types de graphes et de graphes déroulants.....	12-1
Options des graphes et de graphes déroulants	12-2
Échelles X et Y multiples sur des graphes et des graphes déroulants	12-2
Lissage pour les graphes et graphes déroulants.....	12-2
Personnalisation de l'apparence des graphes et des graphes déroulants	12-3
Personnalisation des graphes	12-3
Curseurs de graphe.....	12-4
Mise à l'échelle automatique	12-5
Légende de l'échelle d'un graphe.....	12-5
Formatage des axes	12-5
Formatage dynamique des graphes	12-6
Utilisation du rafraîchissement progressif	12-6
Personnalisation des graphes déroulants	12-7
Longueur de l'historique.....	12-7
Mode de mise à jour du graphe déroulant.....	12-7
Courbes superposées comparées aux courbes empilées	12-8
Graphes et graphes XY	12-9
Types de données des graphes à courbe unique	12-10
Graphes multi-courbes.....	12-10
Types de données des graphes XY à courbe unique	12-11
Types de données des graphes XY multi-courbes.....	12-12
Graphes déroulants	12-12
Graphes et graphes déroulants d'intensité	12-13
Représentation des couleurs	12-15
Options du graphe déroulant d'intensité.....	12-16
Options du graphe d'intensité.....	12-16
Graphes numériques	12-16
Masquage des données	12-19
Graphes 3D	12-19
Type de données waveform	12-20
Type de données waveform numérique	12-21

Chapitre 13

Vis Graphisme et son

Utilisation de l'indicateur image.....	13-1
Vis de tracés d'images	13-2
Utilisation du VI Graphe polaire comme sous-VI.....	13-3
Utilisation des VIs Tracer une waveform et Tracer XY comme sous-VIs.....	13-3
Utilisation des VIs Abaque de Smith comme sous-VIs	13-3

VI de fonctions sur les images	13-4
Création et modification des couleurs avec les VIs Fonctions sur les images	13-6
VI Formats graphiques.....	13-7
VI Son	13-8

Chapitre 14

E/S sur fichiers

Opérations de base d'E/S sur fichiers.....	14-1
Choix du format d'E/S sur fichiers.....	14-2
Quand utiliser des fichiers texte.....	14-2
Quand utiliser des fichiers binaires	14-4
Quand utiliser des fichiers journaux	14-4
Utilisation de VIs d'E/S sur fichiers de haut niveau	14-6
Utilisation de VIs et de fonctions d'E/S sur fichiers de bas niveau et avancés.....	14-7
Enregistrement sur disque en continu	14-8
Création de fichiers texte et de fichiers tableur	14-9
Formatage et écriture de données dans des fichiers	14-10
Balayage de données à partir de fichiers.....	14-10
Création de fichiers binaires.....	14-10
Création de fichiers journaux	14-11
Écrire des données waveform dans un fichier.....	14-11
Lecture de données waveform dans des fichiers	14-12
Paramètres dupliqués.....	14-13
Création de fichiers de configuration	14-13
Utilisation de fichiers de paramètres de configuration	14-14
Format de fichier de paramètres de configuration Windows.....	14-15
Enregistrement des données de la face-avant.....	14-16
Journaux de la face-avant automatiques et interactifs	14-17
Affichage interactif des données de la face-avant enregistrées dans un fichier journal	14-17
Effacer un enregistrement.....	14-18
Effacer la liaison vers le fichier journal.....	14-18
Changer la liaison vers le fichier journal.....	14-19
Récupération des données de la face-avant par programmation.....	14-19
Récupération des données de la face-avant au moyen d'un sous-VI	14-19
Spécifier les enregistrements.....	14-20
Récupération des données de la face-avant au moyen des fonctions d'E/S sur fichiers	14-21
Répertoire de données LabVIEW	14-22
Fichier de données de mesures LabVIEW.....	14-22

Chapitre 15

Documenter et imprimer des VIs

Documentation des VIs.....	15-1
Activation de l'historique des révisions des VIs	15-2
Numéros de révision	15-2
Création de descriptions de VIs et d'objets.....	15-3
Impression de la documentation.....	15-3
Enregistrement de la documentation sous forme de fichiers	
texte, HTML ou RTF	15-4
Sélection des formats d'image pour les fichiers HTML.....	15-4
Conventions d'appellation des fichiers image.....	15-5
Création de vos propres fichiers d'aide.....	15-5
Impression des VIs.....	15-6
Imprimer la fenêtre active	15-6
Impression de VIs par programmation.....	15-7
Impression de la face-avant d'un VI après exécution du VI.....	15-7
Utilisation d'un sous-VI pour imprimer des données à partir	
d'un VI de niveau principal.....	15-8
Génération et impression de rapports.....	15-8
Techniques d'impression supplémentaires.....	15-9

Chapitre 16

Personnalisation des VIs

Configuration de l'apparence et du comportement des VIs.....	16-1
Personnalisation des menus	16-2
Création de menus	16-3
Gestion de la sélection des menus.....	16-3

Chapitre 17

Contrôle des VIs par programmation

Caractéristiques du VI Serveur	17-1
Construire des applications du VI Serveur	17-2
Références d'application et de VI	17-3
Édition des paramètres de l'application et du VI.....	17-4
Nœuds de propriété.....	17-4
Nœuds de propriété liés implicitement	17-5
Nœuds de méthode	17-5
Manipulation des propriétés et des méthodes de classe Application	17-5
Manipulation des propriétés et des méthodes de classe VI.....	17-6
Manipulation des propriétés et des méthodes de classe Application et VI	17-7

Charger et appeler les VIs de façon dynamique	17-8
Nœuds d'appel par référence et refnoms de VI de type strict.....	17-8
Édition et exécution de VIs sur des ordinateurs à distance	17-9
Contrôle des objets de la face-avant.....	17-9
Refnoms de commande de type strict et refnoms de commande	
partiellement spécifiés.....	17-10

Chapitre 18

Mise en réseau dans LabVIEW

Choisir une solution parmi E/S sur fichiers, VI Serveur, ActiveX et mise en réseau ...	18-1
LabVIEW comme client et serveur réseau	18-2
Utilisation de la technologie DataSocket	18-2
Spécification d'une URL	18-3
Formats de données supportés par DataSocket.....	18-5
Utilisation de DataSocket sur la face-avant	18-5
Lecture et écriture de données en direct depuis le diagramme	18-7
Ouverture et fermeture de connexions DataSocket	
par programmation.....	18-8
Buffériser des données DataSocket	18-8
Rapport de diagnostic.....	18-9
DataSocket et données variant.....	18-10
Publication de VIs sur le Web	18-11
Options du serveur Web.....	18-11
Création de documents HTML	18-12
Publication d'images de face-avant	18-12
Formats des images de face-avant	18-13
Affichage et contrôle des faces-avant à distance.....	18-13
Configuration du serveur pour les clients	18-13
Licence pour face-avant distante	18-14
Affichage et contrôle de faces-avant dans LabVIEW ou à partir d'un	
navigateur Web	18-14
Affichage et contrôle des faces-avant dans LabVIEW.....	18-15
Affichage et contrôle de faces-avant à partir d'un	
navigateur Web.....	18-15
Fonctionnalités non supportées dans la visualisation et le contrôle	
des faces-avant distantes	18-16
Envoi de données par e-mail à partir de VIs	18-18
Sélection d'un jeu de caractères.....	18-18
Jeu de caractères US-ASCII	18-19
Jeu de caractères ISO Latin-1	18-19
Jeu de caractères Mac OS	18-20
Translittération.....	18-20

Applications de communication de bas niveau.....	18-21
TCP et UDP.....	18-21
Apple Events et PPC Toolbox (Mac OS).....	18-22
VIs de canaux (UNIX).....	18-22
Exécution des commandes système (Windows et UNIX).....	18-22

Chapitre 19

Connectivité Windows

Environnement .NET.....	19-2
Fonctions et nœuds .NET.....	19-3
LabVIEW comme client .NET.....	19-4
Mappage des types de données.....	19-5
Déploiement d'applications .NET.....	19-5
Déploiement d'un exécutable.....	19-5
Déploiement de VIs.....	19-6
Déploiement de DLLs.....	19-6
Configuration d'une application client .NET (avancée).....	19-6
Objets, propriétés, méthodes et événements ActiveX.....	19-6
VIs, fonctions, commandes et indicateurs ActiveX.....	19-7
LabVIEW en tant que client ActiveX.....	19-8
Accès à une application pour laquelle ActiveX est activé.....	19-8
Insertion d'un objet ActiveX sur la face-avant.....	19-9
Mode de conception des objets ActiveX.....	19-9
Définition des propriétés ActiveX.....	19-10
Navigateur de propriétés ActiveX.....	19-10
Pages de propriétés ActiveX.....	19-10
Nœuds de propriété.....	19-11
LabVIEW en tant que serveur ActiveX.....	19-12
Support pour les interfaces ActiveX Automation personnalisées.....	19-12
Utilisation de constantes pour définir des paramètres dans les VIs ActiveX.....	19-12
Événements ActiveX.....	19-14
Gestion d'événements ActiveX.....	19-14

Chapitre 20

Appel de code provenant de langages de programmation textuels

Appeler une fonction d'une DLL.....	20-1
Code Interface Node.....	20-2

Chapitre 21

Formules et équations

Méthodes d'utilisation d'équations dans LabVIEW	21-1
Boîtes de calcul.....	21-2
Utilisation de la boîte de calcul.....	21-2
Variables dans la boîte de calcul.....	21-3
Nœuds d'expression	21-4
Polymorphisme dans les nœuds d'expression.....	21-5
Nœuds de script MATLAB	21-5
Suggestions concernant la programmation des scripts MATLAB.....	21-6

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
Mac OS	A-2
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-3
Mode Comparer des ensembles	C-3
Clusters	C-3
Mode Comparer des éléments	C-3
Mode Comparer des ensembles	C-4

Annexe D

Support technique et services professionnels

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, notamment l'interface utilisateur LabVIEW et l'environnement de programmation, ainsi que les palettes et les outils LabVIEW. Ce manuel n'inclut pas les informations spécifiques à chaque palette, outil, menu, boîte de dialogue, commande ou encore à chaque fonction ou VI intégré. 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 obtenir 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 (Portable Document Format). 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.



Remarque Vous devez avoir installé Adobe Acrobat Reader avec fonctions de recherche et d'accessibilité version 5.0.5 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 à la section *Ressources de documentation LabVIEW* du chapitre 1, *Introduction à LabVIEW*, pour obtenir de plus amples informations sur l'accès aux fichiers PDF dans la *Bibliothèque LabVIEW*.

Organisation de ce manuel

Le *Manuel de l'utilisateur LabVIEW* comprend deux sections. Partie I, *Concepts LabVIEW*, décrit les concepts de programmation pour la construction d'applications dans LabVIEW. Les chapitres de cette section 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 invite à cliquer sur le menu déroulant **Fichier**, à sélectionner l'élément **Mise en page**, puis **Options** dans la dernière boîte de dialogue.



Cette icône signale une astuce qui vous fournit 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 indique aussi un texte qui est substituable par un mot ou une valeur que vous devez fournir.

monospace	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 aussi utilisée pour les noms propres d'unités de disque, de chemins, de répertoires, de programmes, de sous-programmes, de périphériques, de fonctions, d'opérations, de variables, de fichiers et d'extensions et d'extraits de code.
gras monospace	Le texte en gras de cette police signale les messages et réponses que l'ordinateur affiche automatiquement à l'écran. Ceci accentue également les lignes de code 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	(Mac OS) Appuyez sur <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 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 complète qui l'accompagne, 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 LabVIEW que vous utilisez pour construire les faces-avant et les diagrammes des VIs. Ce chapitre décrit aussi comment personnaliser les palettes LabVIEW et définir plusieurs options de l'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 au lieu de 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.

Dans LabVIEW, vous construisez une interface utilisateur à l'aide d'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 commander 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 kits d'outils logiciels (toolsets) supplémentaires 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 obtenir des informations complémentaires 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 avoir installé Adobe Acrobat Reader avec fonctions de recherche et d'accessibilité version 5.0.5 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 de manuels des produits National Instruments à l'adresse ni.com/manuals pour vous procurer 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 la Bibliothèque LabVIEW**.
- **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.

- **Carte de référence rapide LabVIEW** : utilisez cette carte comme référence en vue d'obtenir des informations sur les ressources en aide, les raccourcis clavier, les types de données des terminaux et les outils d'édition, d'exécution 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* comprend aussi des instructions très détaillées pour utiliser les fonctionnalités de LabVIEW. Accédez à l'*Aide LabVIEW* en sélectionnant **Aide»Aide LabVIEW...**

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

- *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 NI Developer Zone, la base de connaissances (KnowledgeBase) et la bibliothèque des manuels sur les produits



Remarque (Mac OS et UNIX) National Instruments vous recommande d'utiliser Netscape 6.0 ou une version ultérieure, ou bien Internet Explorer 5.0 ou une version ultérieure pour afficher 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.
- **Guide de l'utilisateur pour l'Application Builder de LabVIEW** : utilisez ce document pour vous familiariser avec LabVIEW Application Builder, compris dans le Système de développement professionnel LabVIEW et que vous pouvez acheter séparément.

Ce guide de l'utilisateur comprend des instructions sur l'installation d'Application Builder, décrit la configuration système requise pour les applications, ainsi que les modifications qui ont été apportées entre les versions précédentes et la version actuelle. Ce guide de l'utilisateur décrit également les mises en garde et les recommandations dont vous devez tenir compte lorsque vous créez un VI dans une application ou une bibliothèque partagée.

- **LabVIEW Development Guidelines** : utilisez ce manuel pour apprendre à construire des VIs faciles à comprendre, à utiliser et à modifier. Ce manuel décrit des techniques de suivi, de conception et de documentation de projet. Ce manuel contient en outre des directives concernant le style recommandé.



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.

- **LabVIEW Analysis Concepts** : utilisez ce manuel pour apprendre quels sont les concepts d'analyse utilisés par LabVIEW. Ce manuel comprend des informations sur la génération de signaux, la transformée de Fourier rapide (FFT) et la transformée de Fourier discrète (DFT), les fenêtres de lissage, l'ajustement de courbe, l'algèbre linéaire, les concepts fondamentaux de probabilités et de statistiques, ainsi que sur l'analyse point par point pour une analyse en temps réel.



Remarque Le manuel *LabVIEW Analysis Concepts* est disponible uniquement au format PDF.

- **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 comprend des informations sur l'appel à des bibliothèques partagées (DLLs), les bibliothèques de fonctions, les programmes de manipulation de mémoire et de fichiers, et les programmes de diagnostics.



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

- **Notes d'informations LabVIEW** : utilisez ces notes d'informations pour installer et désinstaller LabVIEW. Ces notes d'informations décrivent la configuration système nécessaire au logiciel LabVIEW et contiennent des informations sur les problèmes connus avec LabVIEW.
- **Notes de mise à jour LabVIEW** : utilisez ces notes de mise à jour pour mettre à niveau LabVIEW pour Windows, Mac OS et UNIX vers la dernière version. Les notes de mise à jour décrivent aussi les nouvelles fonctionnalités et problèmes que vous pourriez rencontrer lors de la mise à niveau.
- **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/zone 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 est livré avec le matériel VXI. National Instruments vous recommande d'utiliser la technologie VISA pour configurer, programmer et dépanner vos systèmes d'instrumentation basés sur du matériel VXI.



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

VIs modèles, VIs d'exemple et outils LabVIEW

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

VIs modèles LabVIEW

Les VIs modèles LabVIEW comprennent les sous-VIs, fonctions, structures et objets de la face-avant dont vous avez besoin pour commencer à construire des applications de mesure courantes. Les VIs modèles s'ouvrent comme VIs sans titre que vous devez enregistrer. Sélectionnez **Fichier**»**Nouveau** pour afficher la boîte de dialogue **Nouveau** qui comprend les VIs modèles. Vous pouvez aussi afficher la boîte de dialogue **Nouveau** en cliquant sur le bouton **Nouveau** de la boîte de dialogue **LabVIEW**.

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 effectuer des copier-coller à partir d'un ou de plusieurs exemples dans votre propre VI. Parcourez ou recherchez les VIs d'exemple en sélectionnant **Aide»Recherche d'exemples**. Reportez-vous à la page NI Developer Zone à l'adresse ni.com/zone pour trouver d'autres VIs d'exemple.

Outils LabVIEW

LabVIEW comprend de nombreux outils pour vous aider à configurer rapidement vos appareils de mesure. Vous pouvez accéder aux outils suivants à partir du menu **Outils**.

- **(Windows)** Measurement & Automation Explorer (MAX) vous aide à configurer vos logiciels et matériel National Instruments.
- **(Mac OS 9 ou version antérieure)** L'utilitaire de configuration NI-DAQ vous aide à configurer le matériel DAQ de National Instruments.
- **(Mac OS 9 ou version antérieure)** L'Assistant Entrées/Sorties DAQ vous aide à définir le type de périphérique connecté aux voies du matériel DAQ. Une fois que vous avez défini une voie, l'Assistant Entrées/Sorties DAQ se souvient des paramètres.
- **(Mac OS 9 ou version antérieure)** L'afficheur de voies virtuelles répertorie les voies DAQ configurées.
- **(Mac OS 9 ou version antérieure)** L'Assistant Solutions DAQ vous aide à trouver des solutions pour des applications DAQ courantes. Vous pouvez choisir entre des exemples de VIs ou créer des VIs personnalisés.

Utilisez l'Assistant DAQ pour configurer graphiquement des voies ou des tâches de mesure courantes. Vous pouvez accéder à l'Assistant DAQ par les méthodes suivantes :

- Placez le VI Express Assistant DAQ sur le diagramme.
- Cliquez avec le bouton droit sur une commande Voie globale DAQmx et sélectionnez **Nouvelle voie (Assistant DAQ)** dans le menu local. Cliquez avec le bouton droit sur une commande Nom de tâche DAQmx et sélectionnez **Nouvelle tâche (Assistant DAQ)** dans le menu local. Cliquez avec le bouton droit sur une commande Nom de l'échelle DAQmx et sélectionnez **Nouvelle échelle (Assistant DAQ)** dans le menu local.

- Lancez Measurement & Automation Explorer et sélectionnez **Voisinage de données** ou **Échelles** dans l'arbre **Configuration**. Cliquez sur le bouton **Créer un nouvel objet...** Configurez une voie, une tâche ou une échelle NI-DAQmx.

Reportez-vous au *LabVIEW Measurements Manual* pour obtenir des informations complémentaires sur l'utilisation de l'Assistant DAQ.

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 de l'interface utilisateur ou d'autres sources et qui affichent ces informations ou les déplacent 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 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.

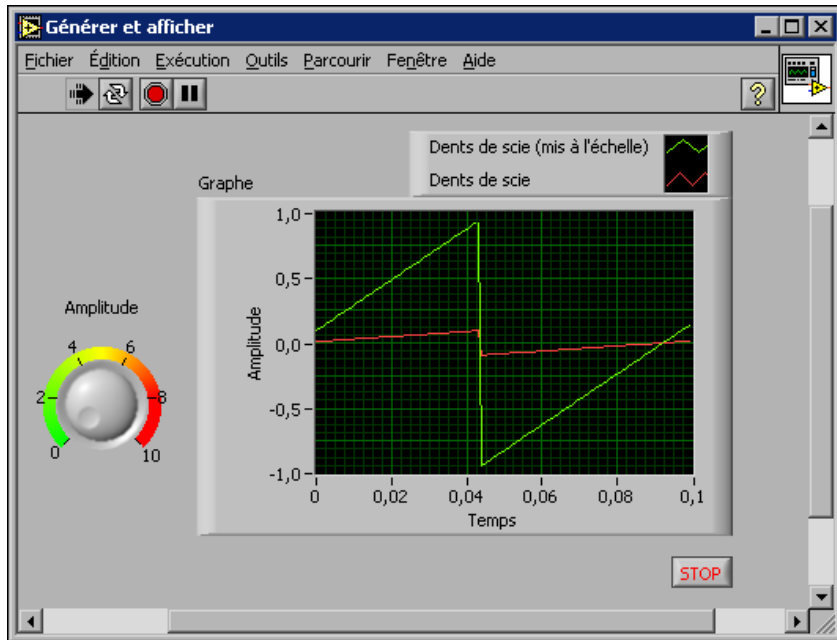


Figure 2-1. Exemple de face-avant

Vous construisez la face-avant avec des commandes et des indicateurs qui sont respectivement les terminaux d'entrée et terminaux de sortie interactifs du VI. 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 types d'affichage. 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. 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 commander 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 obtenir 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.

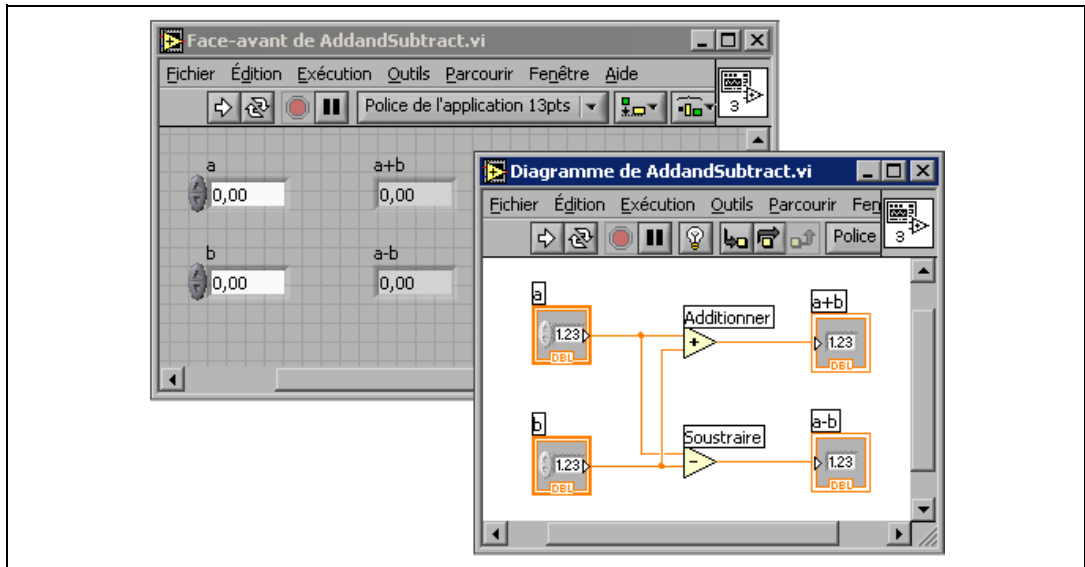


Figure 2-2. Exemple de diagramme et face-avant correspondante

Terminaux



Les terminaux représentent le type de données de la commande ou de l'indicateur. Vous pouvez configurer des commandes ou des indicateurs de la face-avant pour qu'ils s'affichent sous forme d'icônes ou de terminaux de type de données sur le diagramme. Par défaut, les objets de la face-avant sont représentés par des terminaux d'icônes. Par exemple, un terminal d'icône de bouton rotatif indiqué sur la gauche représente un bouton rotatif sur la face-avant. La mention DBL au bas du terminal représente un type de données numérique à virgule flottante double précision. Un terminal DBL, comme celui qui est 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 obtenir 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 Additionner et Soustraire. Lorsque les fonctions Additionner 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 (**a+b** et **a-b** dans la figure 2-2).

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 Additionner 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 obtenir de plus amples informations sur les 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 des commandes et indicateurs aux fonctions Additionner 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 sous la forme d'une ligne noire en pointillé avec un symbole x rouge au milieu. Reportez-vous à la section *Utilisation des fils de liaison pour relier les objets du diagramme* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur les fils de liaison.

Structures

Les structures sont des représentations graphiques de boucles et de conditions dans les langages de programmation textuels. Utilisez des structures dans votre diagramme pour répéter des blocs de code et pour exécuter le code de manière conditionnelle ou dans un ordre spécifique. Reportez-vous au chapitre 8, *Boucles et structures*, pour obtenir des exemples et des informations complémentaires 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 sous-VI. Chaque VI affiche une icône, comme celle qui est 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 les 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 obtenir 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 qui correspond 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 obtenir de plus amples informations sur la configuration des connecteurs.



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.

Utilisation et personnalisation des VIs et sous-VIs

Après avoir construit un VI et créé ses icônes et son connecteur, vous pouvez l'utiliser comme sous-VI. Reportez-vous à la section [Sous-VIs](#) du chapitre 7, [Création de VIs et de sous-VIs](#), pour obtenir 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 obtenir 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 16, [Personnalisation des VIs](#), pour obtenir 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. La palette de **Commandes** contient les commandes et les indicateurs que vous utilisez pour créer 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 obtenir de plus amples informations sur les types de commande et d'indicateur. Les commandes et les indicateurs qui se trouvent sur la palette de **Commandes** dépendent du type de palette actuellement sélectionné. Reportez-vous à la section *Création et édition d'un type de palette personnalisé* de ce chapitre pour obtenir de plus amples informations sur les types de palette.

Sélectionnez **Fenêtre»Palette de commandes** ou effectuez 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. LabVIEW garde en mémoire la position et la taille de la palette de **Commandes** de sorte que lorsque vous redémarrez LabVIEW, la palette s'affiche au même endroit et garde la même dimension.

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 obtenir 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. La palette de **Fonctions** 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 obtenir de plus amples informations sur les types de VIs et de fonctions. Les VIs et les fonctions qui se trouvent sur la palette de **Fonctions** dépendent du type de palette actuellement sélectionné. Reportez-vous à la section [Création et édition d'un type de palette personnalisé](#) de ce chapitre pour obtenir de plus amples informations sur les types de palette.

Sélectionnez **Fenêtre»Palette de fonctions** ou effectuez 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. LabVIEW garde en mémoire la position et la taille de la palette de **Fonctions** de sorte que lorsque vous redémarrez LabVIEW, la palette s'affiche au même endroit et garde la même dimension.

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

Lorsque vous cliquez sur l'icône d'une sous-palette, la palette entière devient la sous-palette que vous avez sélectionnée. Cliquez sur un objet de la palette pour le placer sur le curseur de sorte que vous puissiez placer cet objet sur la face-avant ou le diagramme. 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.

Utilisez les boutons suivants sur les barres d'outils des palettes de **Commandes** et de **Fonctions** pour naviguer et rechercher des commandes, des VIs et des fonctions.



- **Remonter l'arborescence** : vous fait passer à un niveau supérieur dans la hiérarchie de la palette. Cliquez sur ce bouton et maintenez enfoncé le bouton de la souris pour afficher un menu local qui répertorie chaque sous-palette dans le chemin de la sous-palette actuelle. Sélectionnez le nom d'une sous-palette dans le menu local pour naviguer vers cette sous-palette.



- **Rechercher** : fait passer la palette en mode de recherche afin que vous puissiez effectuer des recherches textuelles pour localiser des commandes, des VIs ou des fonctions dans les palettes. Pendant que la palette est en mode de recherche, cliquez sur le bouton **Revenir à la palette** pour quitter le mode de recherche et revenir à la palette.



- **Options** : affiche la page **Palettes de commandes/fonctions** de la boîte de dialogue **Options**, permettant de sélectionner un type de palette ainsi que le format des palettes.



- **Restaurer la taille de palette** : redimensionne la palette à sa taille par défaut. Ce bouton apparaît uniquement si vous redimensionnez la palette de **Commandes** ou de **Fonctions**.

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 la palette d'**Outils** n'importe où sur l'écran. LabVIEW garde en mémoire la position de la palette d'**Outils** de sorte que lorsque vous redémarrez LabVIEW, la palette s'affiche à la même position.



Conseil Appuyez sur la touche <Maj> et effectuez 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**. Vous pouvez désactiver la sélection automatique de l'outil en cliquant sur le bouton **Sélection automatique de l'outil** situé sur la palette d'**Outils**, représentée à gauche. Appuyez sur les touches <Maj-Tab> ou cliquez sur le bouton **Sélection automatique de l'outil** pour activer à nouveau la sélection automatique de l'outil. Vous pouvez également désactiver la sélection automatique de l'outil en sélectionnant manuellement un outil sur la palette d'**Outils**. Appuyez sur la touche <Tab> ou cliquez sur le bouton **Sélection automatique de l'outil** sur la palette d'**Outils** pour activer à nouveau la sélection automatique de l'outil.

Menus et 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 en haut de la fenêtre d'un VI contiennent des éléments communs à d'autres applications, tels que **Ouvrir**, **Enregistrer**, **Copier** et **Coller**, ainsi que d'autres éléments spécifiques à LabVIEW. Certains éléments de menus donnent également des raccourcis clavier.

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

(Windows et UNIX) Par défaut, les menus affichent uniquement les éléments utilisés le plus récemment. Cliquez sur les flèches au bas d'un menu pour afficher tous les éléments. Vous pouvez afficher tous les éléments de menu par défaut en sélectionnant **Outils»Options** et en sélectionnant **Divers** dans le menu déroulant du haut de la fenêtre.



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

Menus locaux

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

Aide contextuelle

La fenêtre **Aide contextuelle** affiche des informations élémentaires sur les objets de LabVIEW lorsque vous déplacez le curseur au-dessus de chaque objet. Les objets comportant des informations d'aide contextuelle comprennent les VIs, les fonctions, les constantes, les structures, les palettes, les propriétés, les méthodes, les événements et les composantes de boîtes de dialogue. Vous pouvez aussi utiliser la fenêtre **Aide contextuelle** pour déterminer à quel endroit précis connecter les fils de liaison à un VI ou une fonction. Reportez-vous à la section *Câblage manuel des objets* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur l'utilisation de l'**Aide contextuelle** pour câbler des objets.

Sélectionnez **Aide»Aide contextuelle** pour afficher la fenêtre d'**Aide contextuelle**. Vous pouvez aussi afficher la fenêtre **Aide contextuelle** en cliquant sur le bouton **Afficher la fenêtre d'Aide contextuelle** sur la barre d'outils, représenté à gauche, ou en appuyant sur les touches <Ctrl-H>. **(Mac OS)** Appuyez sur les touches <Command-H>. **(UNIX)** Appuyez sur les touches <Alt-H>.



Vous pouvez placer la fenêtre d'**Aide contextuelle** n'importe où sur l'écran. La fenêtre d'**Aide contextuelle** se redimensionne pour accommoder la description de chaque objet. Vous pouvez aussi redimensionner la fenêtre d'**Aide contextuelle** pour la définir à sa taille maximale. LabVIEW garde en mémoire la position et la taille de la fenêtre d'**Aide contextuelle** de sorte que lorsque vous redémarrez LabVIEW, la fenêtre s'affiche à la même position et reste à la même taille maximale.

Vous pouvez verrouiller le contenu actuel de la fenêtre d'**Aide contextuelle** de sorte qu'il ne soit pas modifié lorsque vous déplacez le curseur sur différents objets. Sélectionnez **Aide»Aide contextuelle verrouillée** pour verrouiller ou déverrouiller le contenu actuel de la fenêtre d'**Aide contextuelle**. Vous pouvez aussi verrouiller ou déverrouiller le contenu de la fenêtre en cliquant sur le bouton **Verrouillage** de la fenêtre d'**Aide contextuelle**, représenté à gauche, ou en appuyant sur les touches <Ctrl-Maj-L>. **(Mac OS)** Appuyez sur les touches <Command-Maj-L>. **(UNIX)** Appuyez sur les touches <Alt-Maj-L>.





Cliquez sur le bouton **Afficher les terminaux optionnels et le chemin complet** de la fenêtre d'**Aide contextuelle**, représenté à gauche, pour afficher les terminaux optionnels d'un connecteur et afficher le chemin complet d'un VI. Reportez-vous à la section *Définition des entrées et sorties nécessaires, recommandées et facultatives* du chapitre 7, *Création de VIs et de sous-VIs*, pour obtenir des informations complémentaires sur les terminaux optionnels.

Si il existe une rubrique correspondante dans l'*Aide LabVIEW* pour un objet que décrit la fenêtre d'**Aide contextuelle**, un lien bleu [Cliquer ici](#) pour obtenir davantage d'aide s'affiche dans la fenêtre d'**Aide contextuelle**. En outre, le bouton **Davantage d'aide** de la fenêtre d'**Aide contextuelle**, représenté à gauche, est activé. Cliquez sur le lien ou sur le bouton pour afficher l'*Aide LabVIEW* afin d'obtenir de plus amples informations sur l'objet.



Reportez-vous à la section *Création de descriptions de VIs et d'objets* du chapitre 15, *Documenter et imprimer des VIs*, pour obtenir des informations complémentaires sur la création de descriptions à afficher dans la fenêtre d'**Aide contextuelle**.

Personnalisation de votre environnement de travail

Vous pouvez personnaliser les palettes de **Commandes** et de **Fonctions** et utiliser la boîte de dialogue **Options** pour sélectionner un type de palette et définir d'autres options d'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.



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 ou réinstallation. Enregistrez vos VIs et commandes dans le répertoire `user.lib` pour les ajouter aux palettes de **Commandes** et de **Fonctions**.

Ajout de VIs et de commandes aux sous-palettes Utilisateur et Drivers d'instruments

La méthode la plus simple d'ajouter des VIs et des commandes aux palettes de **Fonctions** et de **Commandes** consiste à les enregistrer dans le répertoire `labview\user.lib`. Lorsque vous redémarrez LabVIEW, les palettes **Bibliothèques utilisateur** et **Commandes utilisateur** contiennent des sous-palettes pour chaque répertoire, bibliothèque de VIs (`.lib`) ou fichier de menu (`.mnu`) dans `labview\user.lib` et des icônes pour chaque fichier de `labview\user.lib`. Après que vous avez ajouté ou supprimé des fichiers de répertoires spécifiques, LabVIEW met automatiquement à jour les palettes lors de son redémarrage.

La palette **Drivers d'instruments** correspond au répertoire `labview\instr.lib`. Enregistrez les drivers d'instruments dans ce répertoire pour les ajouter à la palette de **Fonctions**.

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

Création et édition d'un type de palette personnalisé

Pour contrôler le nom de chaque sous-palette et l'emplacement exact des VIs et des commandes que vous ajoutez aux palettes de **Commandes** et de **Fonctions**, vous devez créer un type de palette personnalisé. LabVIEW comprend deux types de palette intégrés : Express et Avancé. Sélectionnez **Outils**»**Avancé**»**Éditer les types de palettes** pour créer ou modifier des types de palette personnalisés.



Remarque Vous ne pouvez pas modifier un type de palette intégré.

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 type que vous avez créé ou installé. Si vous utilisez LabVIEW sur un réseau, vous pouvez définir des répertoires de menus individuels pour chaque utilisateur, ce qui facilite le transfert des types de palette à d'autres personnes.

Lorsque vous créez un nouveau type de palette, LabVIEW utilise une copie du type d'origine intégré, auquel vous pouvez alors apporter 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 le type d'origine.

Comment LabVIEW enregistre les types

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 un type, LabVIEW recherche le sous-répertoire correspondant à ce type 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 le sous-répertoire que LabVIEW crée automatiquement chaque fois que vous créez un type.

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 et des modules dans les palettes

Les toolsets et les modules avec des commandes ou des VIs dans `vi.lib\addons` apparaissent sur les palettes de **Commandes** et de **Fonctions** une fois que vous avez redémarré LabVIEW. Dans le type de palette Express intégré, les toolsets et les modules installent des sous-palettes sur les sous-palettes de **Toutes les commandes** et de **Toutes les fonctions**. Dans le type de palette Avancé intégré, les toolsets et les modules installent des sous-palettes au niveau principal des palettes de **Commandes** et de **Fonctions**.

Si vous installez des commandes et des VIs de toolset ou de module en dehors du répertoire `vi.lib\addons`, vous pouvez déplacer les commandes et les VIs jusqu'au répertoire `vi.lib\addons` pour les ajouter aux palettes.

Paramétrage des options de l'environnement de travail

Sélectionnez **Outils»Options** pour personnaliser LabVIEW. Utilisez la boîte de dialogue **Options** pour définir les options pour les faces-avant, les diagrammes, les chemins, les performances et les problèmes de disque, la grille d'alignement, les palettes, l'opération d'annulation, les outils de mise au point, les couleurs, les polices, l'impression, la fenêtre d'**Historique** 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 du nom de l'option et de 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**.

Mac OS

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 par double double, 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 avec des commandes et des indicateurs qui sont respectivement les terminaux d'entrée et terminaux de sortie 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 types d'affichage. 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.

Sélectionnez **Fenêtre»Palette de commandes** pour afficher la palette de **Commandes**, puis sélectionnez les commandes et indicateurs sur la palette **Commandes** 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

Utilisez les boîtes de dialogue de propriétés ou les menus locaux pour configurer le mode d'affichage ou le comportement des commandes et des indicateurs sur la face-avant. Utilisez les boîtes de dialogue de propriétés lorsque vous souhaitez configurer une commande ou un indicateur de la face-avant par l'intermédiaire d'une boîte de dialogue comprenant une aide contextuelle ou si vous voulez définir plusieurs propriétés à la fois pour un objet. Utilisez les menus locaux pour configurer rapidement les propriétés courantes d'une commande ou d'un indicateur. Les options qui figurent dans les boîtes de dialogue de propriétés et les menus locaux diffèrent

suisant l'objet de la face-avant. Toute option définie à l'aide d'un menu local est prioritaire sur une option définie au moyen d'une boîte de dialogue de propriétés. Reportez-vous à la note d'application *LabVIEW Custom Controls, Indicators, and Type Definitions* pour obtenir de plus amples informations sur la création et l'utilisation de commandes, d'indicateurs et de définitions de type personnalisées.

Cliquez avec le bouton droit sur une commande ou un indicateur de la face-avant et sélectionnez **Propriétés** dans le menu local pour accéder à la boîte de dialogue de propriétés de cet objet. Vous ne pouvez pas accéder aux boîtes de dialogue de propriétés d'une commande ou d'un indicateur tant qu'un VI est en cours d'exécution.

Afficher et masquer des éléments optionnels

Les commandes et indicateurs de la face-avant possèdent des éléments optionnels que vous pouvez afficher ou masquer. Définissez les éléments visibles d'une commande ou d'un indicateur sur l'onglet **Apparence** de la boîte de dialogue de propriétés de l'objet de la face-avant. Vous pouvez aussi définir les éléments visibles 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 de commandes en indicateurs et inversement

LabVIEW configure initialement les objets de la palette de **Commandes** en tant que commandes ou indicateurs en fonction de leur usage principal. Par exemple, si vous sélectionnez un commutateur, il apparaît sur la face-avant en tant que commande car les commutateurs sont utilisés le plus souvent comme dispositifs 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 **Numérique** contient une commande numérique et un indicateur numérique.

Vous pouvez changer une commande en indicateur en cliquant avec le bouton droit sur l'objet et en sélectionnant **Changer en indicateur** dans le menu local, et changer un indicateur en commande en cliquant avec le bouton droit sur l'objet et en sélectionnant **Changer en commande** 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 que vous 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 **Commandes** temporaire pour remplacer l'objet actuellement sur 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.

Configuration de la face-avant

Vous pouvez personnaliser la face-avant en définissant l'ordre de tabulation des objets de la face-avant, en utilisant des graphiques importés et en définissant les objets de la face-avant de sorte qu'ils se redimensionnent automatiquement lorsque la taille de la fenêtre est modifiée.

Paramétrage des raccourcis clavier des commandes

Vous pouvez affecter des raccourcis clavier aux commandes, afin que les utilisateurs puissent naviguer sur la face-avant sans souris. Effectuez un clic droit sur la commande, puis sélectionnez **Avancé»Raccourci clavier** dans le menu local pour afficher la boîte de dialogue **Raccourci clavier**.



Remarque LabVIEW ne répond pas aux raccourcis clavier associés aux commandes masquées.

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 espace ou sur la touche <Entrée> pour changer sa valeur.

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



Remarque Vous pouvez aussi utiliser l'événement Touche enfoncée pour générer un événement lorsque l'utilisateur appuie sur une touche spécifique du clavier.

Reportez-vous à la section *Key Navigation* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière de définir des raccourcis clavier dans une interface utilisateur.

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 configurer 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.

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

Si vous associez la touche <Entrée> à une commande, l'utilisateur ne pourra entrer de retour chariot dans aucune commande chaîne sur cette face-avant. 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 naviguez vers une commande booléenne et que vous appuyez sur la touche <Entrée>, la commande booléenne change, même si une autre commande utilise la touche <Entrée> comme raccourci clavier. Le raccourci clavier <Entrée> attribué n'est appliqué que lorsqu'aucune commande booléenne n'est sélectionnée.

Définition de l'ordre de défilement par tabulation des objets de la face-avant

Les commandes et indicateurs d'une face-avant ont un ordre, appelé ordre de défilement par tabulation, 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 défilement par tabulation est ajusté automatiquement.

L'ordre de défilement par tabulation détermine l'ordre dans lequel LabVIEW sélectionne les commandes et les indicateurs quand l'utilisateur appuie sur la touche <Tab> alors qu'un VI est en cours d'exécution. L'ordre de défilement par tabulation détermine également l'ordre dans lequel les commandes et les indicateurs apparaissent dans les enregistrements des fichiers journaux que vous créez lorsque vous enregistrez les données de la face-avant. Reportez-vous à la section [Enregistrement des données de la face-avant](#) du chapitre 14, *E/S sur fichiers*, pour obtenir de plus amples informations sur l'enregistrement de données.

Vous pouvez définir l'ordre de défilement par tabulation des objets de la face-avant en sélectionnant **Édition»Définir l'ordre de défilement par tabulation**.

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 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.

Reportez-vous à la section *Colors* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière d'utiliser les couleurs dans la conception d'une interface utilisateur.

Utilisation d'images importées

Vous pouvez importer des graphiques provenant d'autres applications pour les utiliser en tant qu'arrière-plan de face-avant, éléments dans des commandes de menu déroulant et 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.

LabVIEW prend en charge la plupart des formats graphiques standard, y compris les formats BMP, JPEG, GIF animés, MNG, MNG animés et PNG. LabVIEW supporte également la transparence.

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....**



Remarque (Windows et Mac OS) Si vous importez une image en utilisant le copier-coller, l'image perd sa transparence.

Reportez-vous au fichier `exemples\general\controls\custom.llb` pour obtenir des exemples de commandes contenant des graphiques importés. Reportez-vous à la section *Graphics and Custom Controls* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière d'utiliser des graphiques dans la conception d'une interface utilisateur.

Alignement et répartition d'objets

Sélectionnez **Exécution»Activer l'alignement sur la grille de la face-avant** afin d'activer l'alignement sur la grille de la face-avant et aligner les objets au fur et à mesure que vous les placez. Sélectionnez

Exécution»Désactiver l’alignement sur la grille de la face-avant pour désactiver l’alignement sur la grille et pour utiliser la grille visible afin d’aligner manuellement les objets. Vous pouvez aussi appuyer sur les touches <Ctrl-#> pour activer ou désactiver l’alignement sur grille. Sur les claviers français, appuyez sur les touches <Ctrl-’>.

(Mac OS) Appuyez sur les touches <Command-*. **(Sun)** Appuyez sur les touches <Meta-#>. **(Linux)** Appuyez sur les touches <Alt-#>.

Vous pouvez également utiliser l’alignement sur grille dans le diagramme.

Sélectionnez **Outils»Options** et sélectionnez **Grille d’alignement** dans le menu déroulant du haut de la fenêtre pour afficher ou masquer la grille.

Pour aligner des objets après que vous les avez placés, sélectionnez-les et sélectionnez ensuite le menu déroulant **Aligner les objets** sur la barre d’outils. Pour espacer les objets de manière régulière, sélectionnez-les et sélectionnez ensuite le menu déroulant **Répartir les objets** sur la barre d’outils.

Groupement et verrouillage des objets

Utilisez 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 des objets groupés ou verrouillés.

Redimensionnement des objets

Vous pouvez changer la taille de la plupart des objets de la face-avant. Lorsque vous utilisez l’outil Flèche sur un objet redimensionnable, des poignées et des cercles de redimensionnement apparaissent sur l’objet aux points où il peut être redimensionné. 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 objets, comme les boutons rotatifs, 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 de l'objet, appuyez sur la touche <Maj> tout en cliquant sur les poignées et les cercles de redimensionnement pour les faire glisser. Pour redimensionner un objet autour de son point central, appuyez sur la touche <Ctrl> tout en cliquant sur les poignées ou les cercles de redimensionnement pour les faire glisser.

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

Pour redimensionner plusieurs objets de la même taille, sélectionnez-les, puis sélectionnez le menu déroulant **Redimensionner les objets** sur la barre d'outils. Vous pouvez redimensionner tous les objets sélectionnés à la largeur ou à la hauteur du plus grand ou du plus petit objet, et vous pouvez redimensionner tous les objets sélectionnés à une taille spécifique en pixels.

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 regroupez avant tout. Pour mettre un objet à l'échelle, sélectionnez-le 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 lorsque vous exécutez le VI.

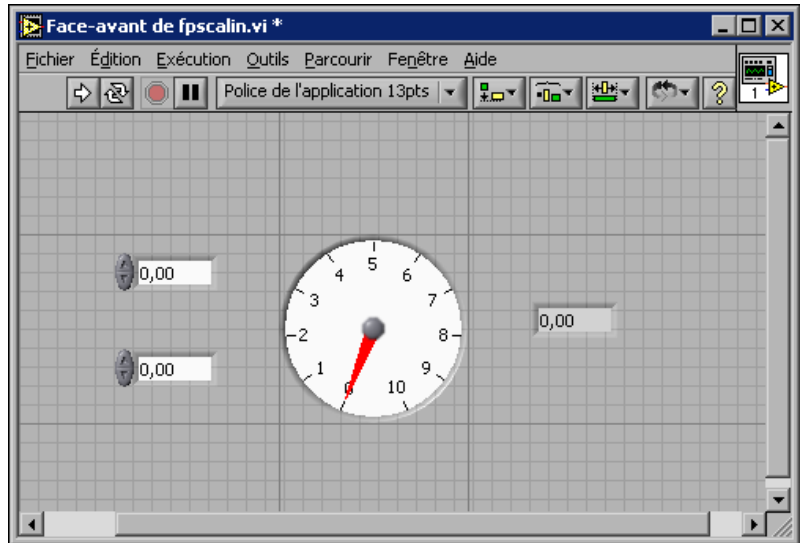


Figure 4-1. Face-avant avec objet mis à l'échelle

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'ajuste 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.

(Mac OS) 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 types d'affichage. 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 classiques et 3D

De nombreux objets de la face-avant ont un aspect tridimensionnel et utilisent un nombre de couleurs élevé. Configurez votre moniteur sur un affichage de couleur d'au moins 16 bits pour obtenir un affichage 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 sous-palette **Commandes classiques** afin de créer des VIs compatibles avec des affichages 256 et 16 couleurs.

Sélectionnez **Fichier»Propriétés du VI** et sélectionnez **Options d'édition** dans le menu déroulant **Catégorie** pour changer le style de commande ou

d'indicateur que LabVIEW crée lorsque vous cliquez avec le bouton droit sur un terminal et que vous sélectionnez **Créer»Commande** ou **Créer»Indicateur** dans le menu local. Reportez-vous à la section *Configuration de l'apparence et du comportement des VIs* du chapitre 16, *Personnalisation des VIs*, pour obtenir de plus amples informations sur la manière de configurer l'apparence et le comportement des VIs. Sélectionnez **Outils»Options** et sélectionnez ensuite **Face-avant** dans le menu déroulant du haut de la fenêtre pour changer le style de commande ou d'indicateur que LabVIEW crée dans les nouveaux VIs lorsque vous effectuez un clic droit sur un terminal et que vous sélectionnez **Créer»Commande** ou **Créer»Indicateur** dans le menu local.

Glissières, boutons rotatifs, cadrans, afficheurs numériques et horodatage

Utilisez les commandes et indicateurs numériques, situés sur les palettes **Numérique** et **Numérique classique** pour simuler des glissières, des boutons rotatifs, des cadrans et des afficheurs numériques. La palette comprend également des boîtes de couleurs et une rampe de couleurs pour définir les valeurs des couleurs, ainsi qu'un paramètre d'horodatage permettant de définir l'heure et la date des données. Utilisez les commandes et indicateurs numériques pour entrer et afficher des données numériques.

Commandes et indicateurs à glissières

Les commandes et indicateurs à glissière comprennent des glissières verticales et horizontales, un réservoir et un thermomètre. Changez la valeur de la commande ou de l'indicateur à glissières 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 et sélectionnez **Ajouter une glissière** dans le menu local pour ajouter d'autres glissières. Le type de données d'une commande à plusieurs glissières est un cluster contenant chacune des valeurs numériques. Reportez-vous à la section *Clusters* du chapitre 10, *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 pratiquement de la même manière que les commandes et les 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 10, *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 constituent la manière la plus simple d'entrer des données numériques et de les afficher. 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 haut ou bas du clavier.

Formatage numérique

Par défaut, LabVIEW affiche et enregistre les nombres de la même manière qu'une calculatrice. Une commande ou un indicateur numérique affiche jusqu'à 6 chiffres avant de passer automatiquement à la notation exponentielle. Vous pouvez configurer le nombre de chiffres que LabVIEW affiche avant qu'il ne passe à la notation exponentielle au moyen de l'onglet **Format et précision** de la boîte de dialogue **Propriétés**.

La précision que vous sélectionnez n'affecte que l'affichage de la valeur. La précision interne dépend toujours de la représentation.

Commande et indicateurs d'horodatage

Utilisez la commande et l'indicateur d'horodatage pour envoyer et récupérer une valeur d'heure et de date en direction et en provenance du diagramme. Vous pouvez changer la valeur de la commande d'horodatage en utilisant les méthodes suivantes :



- Effectuez un clic droit sur la constante et sélectionnez **Format & précision** dans le menu local.
- Cliquez sur le bouton de **navigation Heure/Date**, représenté à gauche, pour afficher la boîte de dialogue **Définir Heure & Date**.
- Sélectionnez **Opérations sur les données»Définir l'heure et la date** dans le menu local pour afficher la boîte de dialogue **Définir Heure & Date**. Vous pouvez aussi cliquer avec le bouton droit sur la commande d'horodatage et sélectionner **Opérations sur les données»Définir l'heure à l'heure actuelle** dans le menu local.

Boîtes de couleur

Une boîte de couleurs affiche une couleur correspondant à une valeur spécifiée. Par exemple, vous pouvez utiliser des boîtes de couleurs pour spécifier différentes conditions, comme les valeurs hors gamme. La valeur de la couleur est exprimée par un nombre hexadécimal sous 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 commandent la valeur de la couleur bleue.

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 la couleur pour afficher sa valeur numérique. Vous configurez une rampe de couleur qui consiste en au moins deux repères arbitraires qui ont chacun 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 12, [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 éléments 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 une rampe de couleurs visible par défaut.

Graphes et graphes déroulants

Utilisez les graphes et les graphes déroulants pour afficher des tracés de données sous forme de graphiques.

Reportez-vous au chapitre 12, *Graphes et graphes déroulants*, pour obtenir des informations complémentaires sur l'utilisation des graphes et des graphes déroulants dans LabVIEW.

Boutons, commutateurs et voyants

Utilisez les commandes et les indicateurs booléens 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 les indicateurs de chaîne et de chemin pour simuler des boîtes et des étiquettes de saisie de texte et pour entrer ou renvoyer 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.

Reportez-vous à la section *Chaînes sur la face-avant* du chapitre 10, *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 de type chaîne déroulante

Utilisez la commande chaîne déroulante pour créer une liste de chaînes que vous pouvez parcourir sur la face-avant. La commande chaîne déroulante est semblable à la commande texte ou menu déroulant. Toutefois, la valeur et le type de données d'une commande chaîne déroulante sont des chaînes au lieu d'être des nombres, comme c'est le cas dans les commandes de menu déroulant. Reportez-vous à la section *Commandes et indicateurs de type énumération et menu déroulant* de ce chapitre pour obtenir de plus amples informations sur les commandes de menu déroulant.

Vous pouvez utiliser une commande chaîne déroulante pour sélectionner les conditions d'une structure Condition. Reportez-vous à la section *Structures Condition* du Chapitre 8, *Boucles et structures*, pour obtenir de plus amples informations sur les structures Condition.

Cliquez avec le bouton droit sur une commande chaîne déroulante et sélectionnez **Éditer les éléments** dans le menu local pour ajouter des chaînes à la liste parmi lesquelles vous pourrez faire vos sélections dans la commande. L'ordre des chaînes dans la page **Éditer les éléments** de la boîte de dialogue **Propriétés de Chaîne déroulante** détermine l'ordre dans lequel sont classées les chaînes dans la commande. Par défaut, la commande chaîne déroulante autorise les utilisateurs à entrer des valeurs de chaînes ne figurant pas déjà dans la liste des chaînes définies pour la commande. Effectuez un clic droit sur la commande chaîne déroulante et sélectionnez **Autoriser les chaînes non définies** dans le menu local afin de désélectionner l'option et d'empêcher l'utilisateur d'entrer des valeurs de chaînes non définies dans la commande.

Lorsque vous tapez une chaîne dans une commande chaîne déroulante lors de l'exécution, LabVIEW sélectionne la première chaîne la plus courte de la commande qui commence par les lettres que vous tapez. Si aucune chaîne ne correspond aux lettres que vous tapez et que la commande n'autorise pas les valeurs de chaîne non définies, LabVIEW n'accepte pas et n'affiche pas les lettres que vous tapez dans la commande.

Lorsque vous configurez la liste de chaînes pour une commande chaîne déroulante, vous pouvez spécifier une valeur personnalisée pour chaque chaîne, ce qui est utilisé si vous voulez que la chaîne qui apparaît dans la commande chaîne déroulante sur la face-avant ne soit pas la même que la valeur de chaîne renvoyée par le terminal chaîne déroulante du diagramme. Effectuez un clic droit sur la commande chaîne déroulante, sélectionnez **Éditer les éléments** dans le menu local et décochez la case **Les valeurs correspondent aux étiquettes** dans la page **Éditer les éléments** de la boîte de dialogue **Propriétés de Chaîne déroulante**. Dans la colonne **Valeurs** de la table qui figure dans cette boîte de dialogue, changez la valeur qui correspond à chaque chaîne dans la commande.

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 chemin, de manière à pouvoir détecter le moment où l'utilisateur ne fournit pas de 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 Mac OS, et sous forme de 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.

(Mac OS) 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 tableau et cluster situés sur les palettes **Tableau et Cluster** et **Tableau et cluster classiques** pour créer des tableaux et des clusters d'autres commandes ou indicateurs. Reportez-vous à la section *Groupement des données avec des tableaux et des clusters* du Chapitre 10, *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 la commande variant. 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 de type liste déroulante, de type arbre et tables

Utilisez les commandes de liste déroulante, situées sur les palettes **Liste et table** et **Liste et table classiques**, pour fournir aux utilisateurs une liste d'éléments parmi lesquels ils peuvent effectuer une sélection.

Listes déroulantes

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.

Lorsque vous tapez des caractères dans une liste déroulante lors de l'exécution, LabVIEW sélectionne le premier élément de la liste déroulante qui commence par les caractères que vous tapez. Utilisez les touches fléchées vers la droite et vers la gauche pour atteindre les éléments suivants et précédents qui correspondent aux caractères que vous avez saisis.

Vous pouvez ajouter un symbole à côté d'un élément de liste, comme dans la boîte de dialogue **Gestionnaire de bibliothèques de VIs**, où les répertoires et les fichiers ont des symboles différents. Vous pouvez aussi insérer des lignes pour séparer les éléments d'une liste.

Utilisez un Nœud de propriété pour modifier des éléments d'une liste et pour recueillir des informations sur les éléments de la liste, comme par exemple, pour détecter les éléments actuellement sélectionnés ou pour

détecter les éléments sur lesquels l'utilisateur a double-cliqué, le cas échéant. Reportez-vous à la section *Nœuds de propriété* du chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur les nœuds de propriété.

Commandes arbre

Utilisez la commande arbre pour fournir aux utilisateurs une liste hiérarchique des éléments parmi lesquels ils peuvent faire leur sélection. Vous organisez les éléments que vous entrez dans la commande arbre par groupes d'éléments ou par nœuds. Cliquez sur le symbole d'expansion contre un nœud pour le développer et afficher tous ses éléments. De même, cliquez sur le symbole contre le nœud pour réduire celui-ci.



Remarque Vous pouvez créer et éditer des commandes arbre uniquement dans les systèmes de développement complet (Full Development System) et de développement professionnel (Professional Development System) LabVIEW. Si un VI contient une commande arbre, vous pouvez exécuter le VI dans toutes les versions LabVIEW, mais vous ne pouvez pas configurer cette commande dans la version de base.

Lorsque vous tapez des caractères dans une commande arbre lors de l'exécution, LabVIEW sélectionne le premier élément de la commandes arbre qui commence par les caractères que vous tapez. Vous pouvez changer la hiérarchie des éléments dans la commande arbre en sélectionnant un élément et en appuyant sur la touche du point (.) pour indenter l'élément actuel ou sur la touche de la virgule (,) pour déplacer l'élément actuel vers la gauche.

Vous configurez les éléments d'une commande arbre de la même façon que vous configurez ceux d'une liste déroulante. Vous pouvez aussi changer le type de symbole qui apparaît contre chaque nœud et définir si l'utilisateur peut ou non déplacer les éléments au sein de la commande arbre.

Vous pouvez utiliser un Nœud de méthode pour modifier les éléments d'une commande arbre et pour recueillir des informations sur ces éléments, comme par exemple, pour détecter sur quels éléments l'utilisateur a éventuellement double-cliqué. Lorsque vous ajoutez un élément à la commande arbre, LabVIEW crée un tag unique pour cet élément. Ce tag sert à modifier des éléments ou à recueillir des informations sur des éléments par programmation. Cliquez avec le bouton droit sur la commande arbre et sélectionnez **Éditer les éléments** dans le menu local pour modifier les tags que LabVIEW crée pour chaque élément.

Reportez-vous à la section *Nœuds de méthode* du chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur les nœuds de méthode.

Reportez-vous au VI Directory Hierarchy in Tree Control dans `examples\general\controls\Directory Tree Control.llb` pour obtenir un exemple d'utilisation de la commande arbre.

Tables

Utilisez la commande de table, figurant sur les palettes **Liste et table** et **Liste et table classiques**, pour créer une table sur la face-avant.

Reportez-vous à la section *Tables* du chapitre 10, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur l'utilisation des commandes de type table.

Commandes et indicateurs de type énumération et menu déroulant

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

Commandes de 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.

Cliquez avec le bouton droit sur une commande de menu déroulant et sélectionnez **Éditer les éléments** dans le menu local pour ajouter des éléments à la liste parmi lesquels vous pourrez faire vos sélections dans la commande. L'ordre des éléments dans la page **Éditer les éléments** de la boîte de dialogue **Propriétés de Menu déroulant** détermine l'ordre dans lequel sont classés les éléments dans la commande. Vous pouvez aussi configurer la commande de menu déroulant de sorte que les utilisateurs puissent entrer des valeurs numériques qui ne sont pas déjà associés à d'autres entrées dans la liste d'éléments définis pour la commande en

effectuant un clic droit sur la commande de menu déroulant et en sélectionnant **Autoriser les valeurs non définies** dans le menu local.

Pour entrer une valeur non définie dans la commande de menu déroulant lors de l'exécution, cliquez sur la commande, sélectionnez **<Autre>** dans le menu local, entrez une valeur numérique dans l'affichage digital qui apparaît et appuyez sur la touche **<Entrée>**. La valeur non définie s'affiche entre crochets obliques dans la commande de menu déroulant. LabVIEW n'ajoute pas la valeur non définie à la liste d'éléments parmi lesquels vous pouvez faire votre sélection dans la commande.

Lorsque vous configurez la liste des éléments pour une commande de menu déroulant, vous pouvez attribuer une valeur numérique spécifique à chaque élément. Si vous n'attribuez pas de valeurs numériques spécifiques aux éléments, LabVIEW attribue des valeurs séquentielles qui correspondent à l'ordre des éléments de la liste, en commençant avec la valeur 0 pour le premier élément. Pour attribuer des valeurs numériques spécifiques, effectuez un clic droit sur la commande de menu déroulant, sélectionnez **Éditer les éléments** dans le menu local et décochez la case **Valeurs séquentielles** dans la page **Éditer les éléments** de la boîte de dialogue **Propriétés de Menu déroulant**. Dans la colonne **Valeurs** de la table qui figure dans cette boîte de dialogue, changez les valeurs numériques qui correspondent à chaque élément dans la commande. Chaque élément de la commande menu déroulant doit posséder une valeur numérique unique.

Commandes de type énumération

Utilisez les commandes de type énumération pour proposer aux utilisateurs une liste d'éléments parmi lesquels ils peuvent faire leur sélection. Une commande de type énumération est similaire à une commande de texte ou de menu déroulant. Toutefois, le type de données d'une commande de type énumération comprend des informations sur les valeurs numériques et les étiquettes de la commande. Le type de données d'une commande de type énumération est numérique.



Remarque Vous ne pouvez pas autoriser l'utilisateur à entrer des valeurs non définies dans des commandes de type énumération, et vous ne pouvez pas non plus attribuer de valeurs numériques spécifiques aux éléments des commandes de type énumération. Si vous avez besoin de cette fonctionnalité, utilisez plutôt une commande de menu déroulant. Reportez-vous à la section *Commandes de menu déroulant* de ce chapitre pour obtenir de plus amples informations sur les commandes de menu déroulant.

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 et structures*, pour obtenir de plus amples informations sur les structures *Condition*.

La représentation numérique d'une commande de type énumération est un entier non signé 8 bits, 16 bits ou 32 bits. Effectuez un clic droit sur une commande de type énumération et sélectionnez **Représentation** dans le menu local pour modifier la représentation 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 entier non signé. *Incrémenter* incrémente de la dernière valeur énumérée à la première et *Décrémenter* décrémente de la première valeur énumérée à la dernière. Lorsqu'un entier signé est contraint à un 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 câblez une valeur à virgule flottante à un indicateur de type énumération, LabVIEW contraint la valeur à virgule flottante à la valeur numérique la plus proche dans l'indicateur de type énumération. LabVIEW manipule les nombres hors gamme, comme décrit auparavant. Si vous câblez une commande de type énumération à n'importe quelle valeur numérique, LabVIEW contraint la valeur de type énumération à une valeur numérique. Pour câbler une commande de type énumération à un indicateur de type énumération, il faut que les éléments de l'indicateur correspondent à ceux de la commande. Néanmoins, l'indicateur peut avoir plus d'éléments que la commande.

Commandes de type Conteneur

Utilisez les commandes de type conteneur situées sur les palettes **Conteneurs** et **Conteneurs classiques** pour grouper des commandes et des indicateurs, pour afficher la face-avant d'un autre VI sur la face-avant du VI actuel ou (**Windows**) pour afficher des objets ActiveX sur la face-avant. Reportez-vous au chapitre 19, *Connectivité Windows*, de ce manuel pour obtenir de plus amples informations à propos de ActiveX.

Commandes onglet

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.

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 par défaut une commande de type énumération. 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.

Commandes de face-avant secondaire

Utilisez la commande de face-avant secondaire pour afficher la face-avant d'un autre VI sur la face-avant du VI actuel. Par exemple, vous pouvez utiliser une commande de face-avant secondaire pour réaliser une interface utilisateur qui se comporte comme un Assistant. Placez les boutons **Précédent** et **Suivant** sur la face-avant du VI de niveau principal et utilisez une commande de face-avant secondaire pour charger différentes faces-avant pour chaque étape de l'Assistant.



Remarque Vous pouvez créer et éditer des commandes de face-avant secondaire uniquement dans les systèmes de développement complet (Full Development System) et de développement professionnel (Professional Development System) LabVIEW. Si un VI contient une commande de face-avant secondaire, vous pouvez exécuter le VI dans toutes les versions LabVIEW, mais vous ne pouvez pas configurer cette commande dans la version de base.

Quand vous placez une commande de face-avant secondaire sur la face-avant, LabVIEW ne crée pas de terminal de face-avant dans le diagramme pour cette commande. Au lieu de cela, LabVIEW crée un Nœud de méthode sur le diagramme avec la méthode Insérer un VI

sélectionnée. Pour charger un VI dans la commande de face-avant secondaire, câblez au nœud de méthode une référence à ce VI. Reportez-vous au chapitre 17, *Contrôle des VIs par programmation*, de ce manuel pour obtenir des informations complémentaires sur l'utilisation des références aux VIs et sur les nœuds de méthodes.



Remarque Comme la commande de face-avant secondaire n'a pas de terminal, vous ne pouvez pas créer un tableau de commandes de face-avant secondaire, ni créer de définition de type pour une commande de face-avant secondaire. Vous pouvez placer une commande de face-avant secondaire dans un cluster pour la grouper avec d'autres commandes, mais ce cluster ne peut pas contenir une ou plusieurs commandes de face-avant secondaire uniquement.

Si la face-avant du VI que vous voulez charger est ouverte ou si vous l'avez chargée dans une autre commande de face-avant secondaire sur la même face-avant, LabVIEW renvoie une erreur et vous ne pouvez pas charger la face-avant dans la commande de face-avant secondaire. De même, vous ne pouvez pas charger la face-avant d'un VI dans une instance distante de LabVIEW, ni charger des faces-avant de façon récursive.

Vous ne pouvez pas utiliser de raccourcis clavier pour naviguer ou pour commander la face-avant dans la commande de face-avant secondaire.

Si vous chargez un VI qui n'est pas en cours d'exécution, le VI se charge dans la commande de face-avant secondaire en mode d'exécution.

LabVIEW affiche uniquement la zone visible de la face-avant du VI que vous chargez dans la commande de face-avant secondaire. Après que vous avez arrêté le VI contenant la commande de face-avant secondaire, LabVIEW efface sa face-avant dans la commande de face-avant secondaire. Vous pouvez aussi utiliser la méthode Supprimer le VI pour décharger le VI de la commande de face-avant secondaire.

Reportez-vous à `exemples\general\controls\subpanel.llb` pour obtenir des exemples d'utilisation des commandes de face-avant secondaire.

Commandes et indicateurs de nom d'E/S

Utilisez les commandes et indicateurs de nom d'E/S pour transmettre des noms de voie DAQ, des noms de ressource VISA et des noms logiques IVI que vous configurez aux VIs d'E/S pour communiquer avec un instrument ou un périphérique DAQ.

Les constantes de noms d'E/S se trouvent sur la palette de **Fonctions**.



Remarque Toutes les commandes ou constantes de noms d'E/S sont disponibles sur toutes les plates-formes. Cela vous permet de développer des VIs d'E/S sur n'importe quelle plate-forme capable de communiquer avec des périphériques spécifiques à des plates-formes. Cependant, si vous tentez d'exécuter un VI avec une commande d'E/S spécifique à une plate-forme sur une plate-forme qui ne prend pas en charge ce périphérique, vous obtiendrez une erreur.

(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.

(Mac OS) 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.

(Mac OS et UNIX) Utilisez les utilitaires de configuration de votre instrument pour configurer les noms de ressource VISA et les noms logiques IVI. Reportez-vous à la documentation de votre instrument pour obtenir plus d'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 pour manipuler les éléments de données individuels d'une waveform. Reportez-vous à la section *Type de données waveform* du chapitre 12, *Graphes et graphes déroulants*, pour obtenir de plus amples informations sur le type de données waveform.

Commande de waveform numérique

Utilisez la commande de waveform numérique pour manipuler les éléments individuels d'une waveform numérique. Reportez-vous à la section *Type de données waveform numérique* du chapitre 12, *Graphes et graphes déroulants*, pour obtenir de plus amples informations sur la commande de waveform numérique.

Commande de données numériques

La commande de données numériques comprend des données numériques agencées sur des lignes et des colonnes. Utilisez la commande de données numériques pour créer des waveforms numériques ou pour afficher des données numériques extraites d'une waveform numérique. Câblez le type

de données waveform numérique à un indicateur de données numériques pour afficher les échantillons et les signaux d'une waveform numérique. La commande de waveform numérique dans la figure 4-2 représente cinq échantillons contenant chacun huit signaux.

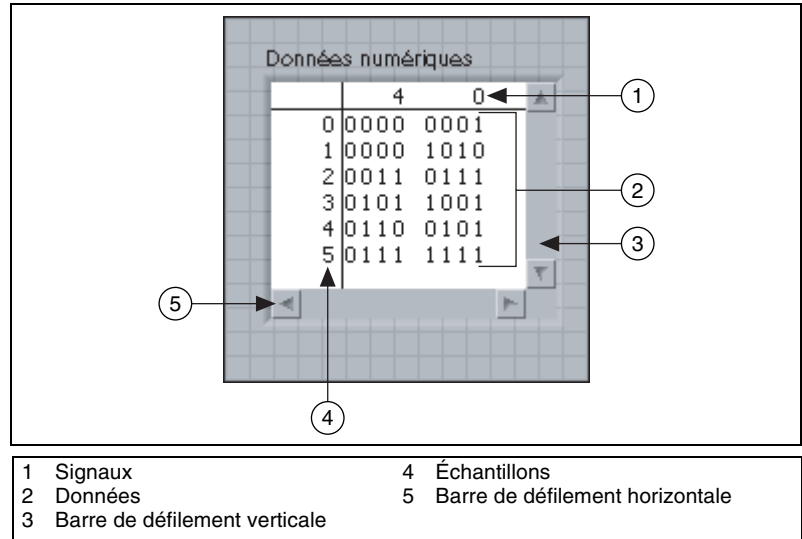


Figure 4-2. Commande de données numériques

Vous pouvez insérer et supprimer des lignes et des colonnes dans la commande de données numériques. Pour insérer une ligne, cliquez avec le bouton droit sur un échantillon dans la colonne des échantillons et sélectionnez **Insérer une ligne avant** dans le menu local. Pour supprimer une ligne, cliquez avec le bouton droit sur un échantillon dans la colonne des échantillons et sélectionnez **Supprimer la ligne** dans le menu local. Pour insérer une colonne, cliquez avec le bouton droit sur un signal dans la colonne des signaux et sélectionnez **Insérer une colonne avant** dans le menu local. Pour supprimer une colonne, cliquez avec le bouton droit sur un signal dans la colonne des signaux et sélectionnez **Supprimer la colonne** dans le menu local.

Vous pouvez aussi couper, copier et coller des données numériques au sein de la commande. Pour couper des données, sélectionnez la ligne ou la colonne que vous voulez couper, cliquez dessus avec le bouton droit et sélectionnez **Opérations sur les données»Couper les données** dans le menu local. Vous ne pouvez couper que des lignes ou des colonnes entières de données. Vous ne pouvez pas créer une nouvelle ligne ou une nouvelle colonne avec les données numériques que vous avez coupées. Pour copier des données, sélectionnez la zone que vous voulez copier, cliquez dessus

avec le bouton droit et sélectionnez **Opérations sur les données»Copier les données** dans le menu local. Pour coller des données, sélectionnez la zone dans laquelle vous voulez copier, cliquez dessus avec le bouton droit et sélectionnez **Opérations sur les données»Coller les données** dans le menu local. Vous devez coller les données numériques dans une zone ayant la même dimension que celle à partir de laquelle vous avez coupé ou copié ces données. Par exemple, si vous copiez quatre bits de données d'une ligne, vous devez sélectionner quatre bits de données existants sur lesquels copier dans la même ligne ou dans une autre ligne. Si vous copiez quatre bits de données à partir d'une zone de deux lignes sur deux colonnes, vous devez coller les données dans une zone de deux lignes sur deux colonnes.

La commande de données numériques et la commande de waveform numérique acceptent les valeurs 0, 1, L, H, Z, X, T et V. Vous pouvez afficher les données dans la commande de données numériques au format binaire, hexadécimal, octal et décimal. Les états numériques L, H, Z, X, T et V, utilisés par certains appareils de mesure, apparaissent sous forme de points d'interrogation si vous optez pour un affichage des données au format hexadécimal, octal ou décimal. Cliquez avec le bouton droit sur la commande, sélectionnez **Format des données** dans le menu local et sélectionnez un format de données pour la commande.

Conversion des données en données numériques

Dans la plupart des cas, le signal que vous acquérez est renvoyé sous forme de données brutes. Pour représenter le signal dans un graphe numérique, vous devez convertir les données brutes que vous avez acquises en données numériques ou en données waveform numérique. Utilisez le VI Waveform analogique en numérique pour convertir les données en données de type waveform numérique. Utilisez la fonction Composantes d'une waveform pour extraire les données numériques des données de type waveform numérique.

Le diagramme de la figure 4-3 simule l'acquisition d'un signal sinusoïdal d'une amplitude de 5, ce qui signifie que le signal sinusoïdal peut contenir des valeurs comprises entre -5 et 5. Le VI Waveform analogique en numérique dans ce diagramme représente chaque valeur par 8 bits. Ces 8 bits peuvent représenter la valeur minimale -5 et la valeur maximale 5. La sonde waveform numérique affiche une portion des valeurs qui résultent sous forme binaire.

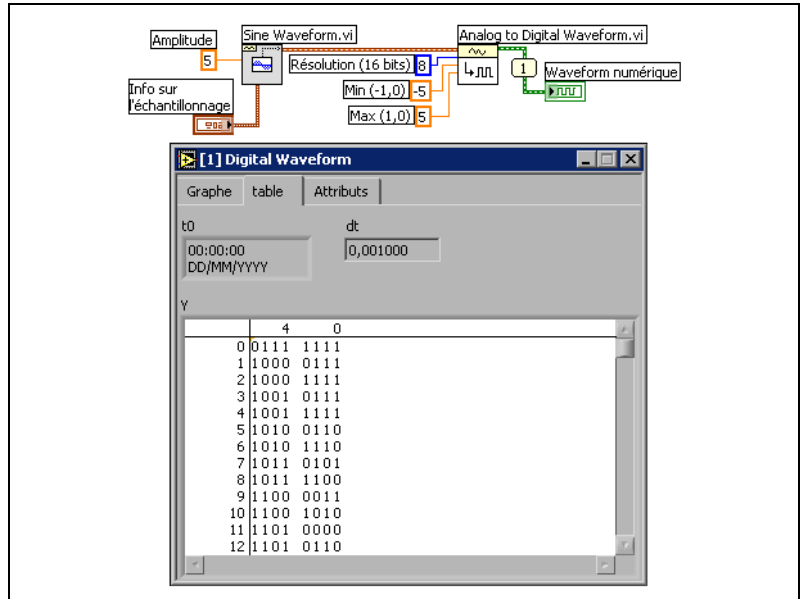


Figure 4-3. Conversion d'une waveform analogique en waveform numérique.

Acquisition d'un sous-ensemble numérique

Utilisez le VI Sous-ensemble d'un signal numérique pour extraire des signaux individuels d'une waveform numérique. Le diagramme de la figure 4-4 vous indique comment combiner des signaux extraits individuellement avec d'autres données numériques pour créer de nouvelles waveforms numériques.

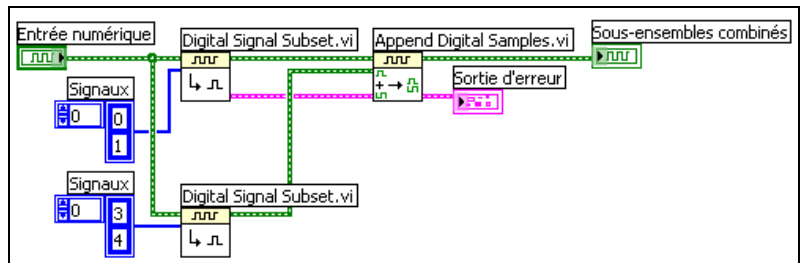


Figure 4-4. Ajout de deux waveforms numériques

Le VI Sous-ensemble d'un signal numérique du haut extrait le premier et le deuxième signal de la waveform numérique. Le VI Sous-ensemble d'un signal numérique du bas extrait le quatrième et le cinquième signal. Le VI Ajouter des signaux numériques ajoute le premier signal au quatrième signal, le deuxième au cinquième, et trace les deux signaux qui résultent sur un graphe numérique.

Ajout d'échantillons et de signaux numériques

Utilisez le VI Ajouter des signaux numériques pour ajouter les échantillons d'un signal numérique à la suite des échantillons d'un autre signal numérique. Vous pouvez ajouter des signaux possédant le même nombre d'échantillons ou des signaux dont le nombre d'échantillons est différent. Par exemple, si vous avez deux signaux numériques contenant chacun deux lignes de 8 bits, les données numériques qui résultent consisteront en deux lignes de 16 bits. Si vous combinez deux signaux, l'un de deux lignes de 8 bits et l'autre d'une ligne de 8 bits, les données numériques qui résultent consisteront en deux lignes de 16 bits. Le VI Ajouter des signaux numériques comble les colonnes restantes dans le second signal par la valeur que vous avez sélectionnée dans l'entrée **valeur par défaut**.

Compression de données numériques

Utilisez le VI Compresser les données numériques pour compresser des données numériques lorsque vous voulez afficher deux signaux série numériques ou plus comportant les mêmes jeux de bits sur la même ligne, afin de voir les données plus clairement. Par exemple, si vous acquérez 10 signaux numériques et que le dixième est différent des neuf autres, le fait de compresser les données numériques vous aide à localiser plus facilement le signal qui est différent. La compression des données numériques économise aussi les ressources en mémoire. Utilisez le VI Décompresser les données numériques pour décompresser les données que vous avez compressées.

Recherche d'un pattern

Utilisez le VI Rechercher une expression numérique pour spécifier le pattern numérique que vous voulez rechercher. Par exemple, si vous acquérez une grande waveform numérique et si vous voulez savoir si une ou plusieurs parties de la waveform correspondent à un certain pattern, câblez ce pattern à l'entrée **pattern numérique** du VI Rechercher une expression numérique afin de détecter toute correspondance.

Références à des objets ou à des applications

Utilisez les commandes et indicateurs de numéro de référence, situés sur les palettes **Refnum** et **Refnum classique**, pour travailler avec des fichiers, des répertoires, des périphériques et des 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 17, *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 à 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 en direction ou en provenance 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.

Étant donné qu'un refnum est une référence temporaire à un objet ouvert, 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 renvoie un refnum différent.

Commandes et indicateurs de boîte de dialogue

Utilisez les commandes de boîte de dialogue dans les boîtes de dialogue que vous créez. Les commandes et les indicateurs de boîte de dialogue sont spécifiquement conçus pour être utilisés dans des boîtes de dialogue et comprennent des commandes de menu déroulant et des commandes numériques pour boîtes de dialogue, des glissières numériques et des barres de progression, des listes déroulantes, des tables, des commandes chemin et chaîne, des commandes à onglet, des commandes arbre, des boutons, des cases à cocher, des boutons radio et une étiquette opaque qui se fond automatiquement à la couleur d'arrière-plan de son parent. 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 système.

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 standard de chaque plate-forme. Sélectionnez **Options d'édition** dans le menu déroulant **Catégorie** pour changer le style de la commande ou de l'indicateur que LabVIEW crée lorsque vous effectuez un clic droit sur un terminal et que vous sélectionnez **Créer»Commande** ou **Créer»Indicateur** dans le menu local. Sélectionnez **Outils»Options** et sélectionnez **Face-avant** dans le menu déroulant du haut de la fenêtre pour changer le style de la commande ou de l'indicateur que LabVIEW crée dans les nouveaux VIs lorsque vous effectuez un clic droit sur un terminal et que vous sélectionnez **Créer»Commande** ou **Créer»Indicateur** dans le menu local. Reportez-vous à la section [Configuration de l'apparence et du comportement des VIs](#) du chapitre 16, [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 ni les supprimer indépendamment de leurs propriétaires. De même, vous pouvez afficher une étiquette d'unité pour les commandes numériques en sélectionnant **Éléments visibles»Étiquette d'unité** dans le menu local.

Reportez-vous à la section *Unités numériques et vérification de type stricte* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur les unités numériques.

Les étiquettes libres ne sont liées à aucun objet et vous pouvez les créer, les déplacer, les faire pivoter ou les supprimer de manière indépendante. Utilisez-les pour annoter les faces-avant et les diagrammes.

Les étiquettes libres sont utiles pour documenter du code sur le diagramme et pour afficher sur la face-avant la liste des instructions à l'intention de l'utilisateur. Double-cliquez sur un espace libre ou utilisez l'outil Texte pour créer des étiquettes libres ou pour éditer l'un ou l'autre type d'étiquettes.

Reportez-vous à la section *Labels* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière de créer des étiquettes descriptives dans une interface utilisateur.

Sous-titres

Les objets de la face-avant peuvent également 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.

Si vous assignez l'objet à un terminal du connecteur, le sous-titre apparaît lorsque vous déplacez le curseur avec l'outil Bobine sur le terminal du diagramme. Le sous-titre apparaît aussi à côté du terminal dans la fenêtre d'**Aide contextuelle**. 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 les connecteurs.

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.

Sélectionnez **Sélection de la police par défaut** dans le 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 sélectionnez **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 de **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 le texte des 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 **Police**, 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.

Reportez-vous à la section *Fonts and Text Characteristics* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière d'utiliser des polices et des caractéristiques de texte dans la conception d'une interface utilisateur.

Conception d'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 et Séquence* du chapitre 8, *Boucles et structures*, 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 la palette **Décorations** pour grouper ou séparer des objets de la face-avant par des boîtes, des lignes ou des flèches. Ces objets, qui ne servent qu'à la décoration, n'affichent pas de 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 empêche 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 la police 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, il la remplace par la couleur 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. Le fait de placer une étiquette ou tout autre objet au-dessus ou partiellement au-dessus d'une commande ou d'un indicateur ralentit les rafraîchissements de l'affichage et peut faire scintiller la commande ou l'indicateur.

Reportez-vous au chapitre 6, *LabVIEW Style Guide*, du manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur l'utilisation de la mise en page, des polices et des couleurs dans la conception d'une interface utilisateur.

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 et indicateurs de boîte de dialogue* de ce chapitre pour obtenir de plus amples informations à propos des commandes des boîtes de dialogue. Reportez-vous à la section *Dialog Boxes* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW*

Development Guidelines pour obtenir de plus amples informations sur la manière d'utiliser les boîtes de dialogue dans la conception d'une interface utilisateur.

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.

Reportez-vous à la section *Sizing and Positioning* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière de sélectionner la taille d'écran d'une interface utilisateur.

Construction du diagramme

Après avoir construit la face-avant, vous devez ajouter le code en utilisant les représentations graphiques des fonctions pour commander 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 sélectionner 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. Durant l'exécution, les données de sortie 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.

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



Vous pouvez configurer des commandes ou des indicateurs de la face-avant pour qu'ils s'affichent sous forme d'icônes ou terminaux de type de données sur le diagramme. Par défaut, les objets de la face-avant sont représentés par des terminaux d'icônes. Par exemple, un terminal icône de bouton rotatif, représenté à gauche, représente un bouton rotatif sur la face-avant. La mention DBL au bas du terminal représente un type de données numérique à virgule flottante double précision. Un terminal DBL, comme celui affiché à gauche, représente une commande ou un indicateur numérique à virgule flottante double précision. Cliquez avec le bouton droit sur un terminal et sélectionnez **Afficher l'icône** dans le menu local pour désélectionner l'option et afficher le type de données de ce terminal. Utilisez des terminaux d'icônes pour afficher sur le diagramme les types d'objets de la face-avant, en plus de leurs types de données. Utilisez les terminaux de types de données pour économiser de l'espace sur le diagramme.



Remarque Les terminaux d'icônes ont une taille plus importante que les terminaux de types de données, c'est pourquoi vous risquez de cacher accidentellement des objets du diagramme lorsque vous convertissez un terminal de type de données en terminal d'icône.

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 VIs et 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 même, des flèches noires apparaissent sur les terminaux de la face-avant pour indiquer si le terminal est une commande

ou un indicateur. Une flèche apparaît sur la droite si le terminal est une commande et sur la gauche si le terminal est un indicateur.

Tableau 5-1. Terminaux des commandes et des indicateurs

































Commande	Indicateur	Type de données	Couleur	Valeurs par défaut
		Numérique à virgule flottante simple précision	Orange	0,0
		Numérique à virgule flottante double précision	Orange	0,0
		Numérique à virgule flottante précision étendue	Orange	0,0
		Numérique complexe à virgule flottante simple précision	Orange	0,0 + i0,0
		Numérique complexe à virgule flottante double précision	Orange	0,0 + i0,0
		Numérique complexe à virgule flottante précision étendue	Orange	0,0 + i0,0
		Numérique entier signé 8 bits	Bleu	0
		Numérique entier signé 16 bits	Bleu	0
		Numérique entier signé 32 bits	Bleu	0
		Numérique entier non signé 8 bits	Bleu	0
		Numérique entier non signé 16 bits	Bleu	0
		Numérique entier non signé 32 bits	Bleu	0
		Horodatage <64.64> bits	Marron	heure et date (locales)
		Type énumération	Bleu	—
		Booléen	Vert	Faux
		Chaîne	Rose	chaîne vide

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

























Commande	Indicateur	Type de données	Couleur	Valeurs par défaut
		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 données de type cluster sont marrons si tous les éléments sont des nombres, et roses dans le cas contraire.	Marron ou rose	—
		Chemin	Turquoise	<Pas un chemin>
		Dynamique	Bleu	—
		Waveform : cluster d'éléments qui comporte les données, instant initial et Δt d'un signal. Reportez-vous à la section <i>Type de données waveform</i> du chapitre 12, <i>Graphes et graphes déroulants</i> , pour obtenir de plus amples informations sur le type de données waveform.	Marron	—
		Waveform numérique	Vert foncé	—
		Données numériques	Vert foncé	—
		Numéro de référence (refnum)	Turquoise	—
		Variant : contient le nom de la commande ou de l'indicateur, les informations du type de données à partir desquelles la conversion s'est effectuée 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 sur le type de données variant.	Violet	—

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

Commande	Indicateur	Type de données	Couleur	Valeurs par défaut
		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 13, <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 sur les fonctions à utiliser avec chaque type de données.

Reportez-vous à la section *Memory and Speed Optimization* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière de sélectionner un type de données en vue d'optimiser l'utilisation de la mémoire.

Constantes

Les constantes sont des terminaux du diagramme qui fournissent des données à valeur fixe au diagramme. Les constantes universelles correspondent à des constantes avec des valeurs fixes, telles que pi (π) et l'infini (∞). Les constantes définies par l'utilisateur sont des constantes que vous définissez et éditez avant l'exécution d'un 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** : ensemble de valeurs physiques et mathématiques de haute précision couramment utilisées, telles que la base du logarithme népérien (e) et la vitesse de la lumière. Les constantes numériques universelles sont situées sur la palette **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 **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 de fichiers universelles se trouvent sur la palette **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, chaîne, tableau et 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. LabVIEW crée une constante qui contient la valeur de la commande de face-avant au moment où vous la faite glisser sur 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 éditer 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 du diagramme qui comportent 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 sur les 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 sur l'utilisation des sous-VIs dans un diagramme.
- **Structures** : effectuent le traitement des éléments de commandes, tels que les structures Séquence déroulée et empilée, structures Condition, boucles For ou boucles While. Reportez-vous au chapitre 8, [Boucles et structures](#), pour obtenir de plus amples informations sur 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 21, [Formules et équations](#), pour obtenir de plus amples informations sur l'utilisation des boîtes de calcul.
- **Nœuds d'expression** : structures permettant de calculer des expressions ou des équations qui contiennent une seule variable. Reportez-vous à la section [Nœuds d'expression](#) du chapitre 21, [Formules et équations](#), pour obtenir de plus amples informations sur l'utilisation des Nœuds d'expression.
- **Nœuds de propriété** : structures qui servent à définir ou à rechercher les propriétés d'une classe. Reportez-vous à la section [Nœuds de propriété](#) du chapitre 17, [Contrôle des VIs par programmation](#), pour obtenir de plus amples informations sur l'utilisation des nœuds de propriété.

- **Nœuds de méthode** : structures permettant d'exécuter les méthodes d'une classe. Reportez-vous à la section *Nœuds de méthode* du chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur l'utilisation des nœuds de méthode.
- **Code Interface Nodes (CIN)** : structures servant à appeler du code écrit dans des langages de programmation textuels. Reportez-vous à la section *Code Interface Node* du chapitre 20, *Appel de code provenant de langages de programmation textuels*, pour obtenir de plus amples informations sur l'appel de code issu de langages de programmation textuels.
- **Nœuds d'appel par référence** : Structures servant à appeler un VI chargé de manière dynamique. Reportez-vous à la section *Nœuds d'appel par référence et refnums de VI de type strict* du chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur l'utilisation des nœuds d'appel par référence.
- **Nœuds d'appel de fonction d'une DLL** : Structures servant à appeler la plupart des bibliothèques partagées standard ou DLLs. Reportez-vous à la section *Appeler une fonction d'une DLL* du chapitre 20, *Appel de code provenant de langages de programmation textuels*, pour obtenir de plus amples informations sur l'utilisation des nœuds d'appel de fonction d'une DLL.

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. Vous ne pouvez ni ouvrir ni éditer une fonction.

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 pour créer et exécuter des opérations arithmétiques, trigonométriques, logarithmiques et mathématiques complexes sur des nombres et pour convertir des nombres d'un type de données à un autre.

Fonctions booléennes

Utilisez les fonctions booléennes pour réaliser des opérations logiques sur des valeurs booléennes uniques ou sur des tableaux de valeurs booléennes, comme par exemple 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 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 numériques en chaînes.
- Formater une chaîne pour l'utiliser avec un traitement de texte ou un tableur.

Reportez-vous à la section *Chaînes* du chapitre 10, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur l'utilisation des fonctions Chaîne.

Fonctions tableau

Utilisez les fonctions Tableau pour créer et manipuler des tableaux, comme dans 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 10, *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 pour créer et manipuler des clusters, comme dans 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 10, *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 pour comparer des valeurs booléennes, des chaînes, des valeurs numériques, des tableaux et des clusters.

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

Fonctions de temps et dialogue

Utilisez les fonctions Temps et Dialogue pour effectuer les tâches suivantes :

- Manipuler la vitesse d'exécution d'une opération.
- Récupérer 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.

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

Fonctions d'E/S sur fichiers

Utilisez les fonctions d'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 dans des fichiers.

La palette **E/S sur fichiers** comprend aussi des VIs qui réalisent des tâches courantes d'E/S sur fichiers. Reportez-vous au chapitre 14, *E/S sur fichiers*, pour obtenir de plus amples informations sur l'utilisation des VIs et des fonctions d'E/S sur fichiers.

Fonctions de waveform

Utilisez les fonctions Waveform pour effectuer les tâches suivantes :

- Construire des waveforms 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 au chapitre 5, *Creating a Typical Measurement Application*, du *LabVIEW Measurements Manual* pour obtenir de plus amples informations sur la création et l'utilisation des données waveform dans les VIs.

Fonctions de contrôle d'applications

Utilisez les fonctions de contrôle d'applications pour commander par programmation les VIs et les applications LabVIEW sur votre ordinateur local ou sur l'ensemble d'un réseau. Reportez-vous au chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur l'utilisation des fonctions de contrôle d'applications.

Fonctions avancées

Utilisez les fonctions avancées pour appeler le code en provenance des bibliothèques, telles que les bibliothèques de liens dynamiques (DLLs), afin de manipuler les données LabVIEW pour les utiliser dans d'autres applications, pour créer et manipuler les clés de registre Windows 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 sur l'utilisation des fonctions avancées.

Ajout de terminaux aux fonctions

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

Vous pouvez ajouter des terminaux aux VIs et fonctions extensibles en utilisant l'outil Flèche pour faire glisser le bord supérieur ou inférieur de la fonction respectivement vers le haut ou le bas. Vous pouvez aussi utiliser l'outil Flèche pour supprimer des terminaux des VIs et 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 la fonction et en sélectionnant **Ajouter une entrée**, **Ajouter une sortie**, **Supprimer une entrée** ou **Supprimer une sortie** dans le menu local. Vous pouvez ajouter des terminaux pour des commandes d'entrée, de sortie ou de 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 à l'aide 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. Vous devez câbler tous les terminaux nécessaires du diagramme. Sinon, le VI est brisé et ne s'exécutera pas. Ouvrez la fenêtre **Aide contextuelle** pour voir quels terminaux sont requis par un nœud du diagramme. Les étiquettes des terminaux nécessaires apparaissent en gras dans la fenêtre **Aide contextuelle**. Reportez-vous à la section *Définition des entrées et sorties nécessaires, recommandées et facultatives* du chapitre 7, *Création de VIs et de sous-VIs*, pour obtenir de plus amples informations sur les terminaux requis. Reportez-vous à la section *Correction des VIs brisés* du chapitre 6, *Exécution et mise au point des VIs*, pour obtenir de plus amples informations sur les VIs brisés.

Les fils de liaison sont de couleurs, de styles et d'épaisseurs différents en fonction de leurs types de données. Un fil de liaison brisé apparaît sous la forme d'un trait noir en pointillés avec un symbole x rouge au milieu. Les flèches de part et d'autre du x rouge indiquent dans quel sens les données sont transmises et la couleur des flèches indique le type de données des données transmises à travers ce fil de liaison. Reportez-vous à la *Carte de référence rapide LabVIEW* pour obtenir de plus amples informations sur les 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 que vous avez 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 segment horizontal ou vertical du fil de liaison. Un coude sur un fil de liaison correspond à la jonction de deux segments. L'endroit auquel deux segments de fil de liaison ou plus 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.

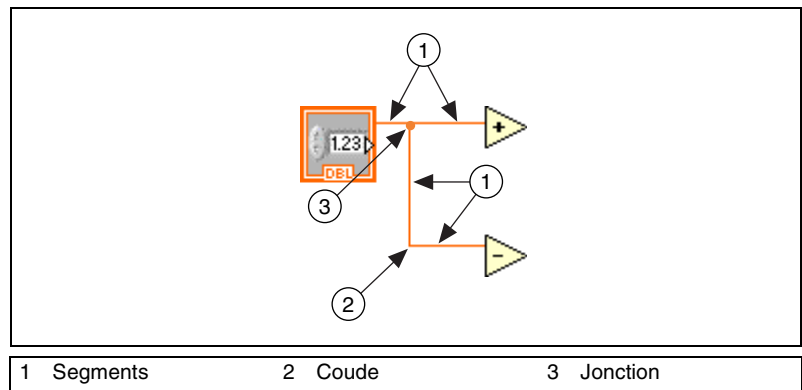


Figure 5-1. Segments de fil de liaison, coude et jonction

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 avec le

bouton de la souris afin de poser le fil, puis déplacez le curseur 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.

(Mac OS) 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.

Reportez-vous à la section *Wiring Techniques* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur les techniques et les astuces de câblage.

Câblage automatique des objets

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 le mieux et ne câble pas ceux qui ne correspondent pas.

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.

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

Vous pouvez désactiver automatiquement le câblage en sélectionnant **Outils»Options** et en sélectionnant **Diagramme** dans le menu déroulant du haut de la fenêtre.

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 terminal, celui-ci clignote. Lorsque vous déplacez l'outil Bobine sur le terminal d'un VI ou d'une fonction, une info-bulle s'affiche également, indiquant le nom du terminal. Si le terminal provoque un fil de liaison brisé, le curseur se change en bobine avec un symbole de mise en garde. Vous pouvez créer le fil de liaison brisé, mais vous devez le corriger avant de pouvoir exécuter le VI. Reportez-vous à la section *Correction des fils de liaison brisés* de ce chapitre pour obtenir de plus amples informations sur la manière de corriger les fils de liaison brisés.

Utilisez la fenêtre **Aide contextuelle** pour déterminer exactement où les fils de liaison doivent être connectés. Lorsque vous déplacez le curseur sur un VI ou sur une fonction, la fenêtre **Aide contextuelle** répertorie chaque terminal de ce VI ou de cette fonction. La fenêtre d'**Aide contextuelle** n'affiche pas les terminaux des VIs ou des fonctions extensibles, comme la fonction Construire un tableau. Cliquez sur le bouton **Afficher les terminaux optionnels et le chemin complet** dans la fenêtre **Aide contextuelle** pour afficher les terminaux optionnels du connecteur. Reportez-vous à la section *Définition des entrées et sorties nécessaires, recommandées et facultatives* du chapitre 7, *Création de VIs et de sous-VIs*, pour obtenir de plus amples informations sur les terminaux optionnels.

Routage des fils de liaison

LabVIEW trouve automatiquement un chemin pour un fil de liaison lors de son câblage. LabVIEW achemine les fils de liaison autour des objets qui se trouvent sur le diagramme, comme les boucles et les structures. LabVIEW guide aussi le routage du fil de liaison de façon à réduire le nombre de coudes dans le câblage. Si possible, les fils de liaison câblés automatiquement à partir de terminaux de commandes sortent du côté droit du terminal et les fils de liaison câblés automatiquement à des terminaux d'indicateurs sortent du côté gauche du terminal.

Pour effectuer un routage automatique d'un fil de liaison existant, cliquez dessus avec le bouton droit de la souris et sélectionnez **Arranger le câblage** dans le menu local.

Appuyez sur la touche <A> après avoir commencé à créer un fil de liaison afin de désactiver le câblage automatique et de router manuellement le fil de liaison. Appuyez à nouveau sur la touche <A> pour activer le routage automatique du fil de liaison. Une fois que vous avez terminé le câblage, LabVIEW réactive le routage automatique des fils de liaison. Vous pouvez également désactiver momentanément le routage automatique après avoir commencé un fil de liaison en maintenant enfoncé le bouton de la souris pendant que vous câblez vers un autre terminal ou un point de liaison, puis en relâchant le bouton de la souris. Dès que vous avez relâché le bouton de la souris, LabVIEW réactive le routage automatique des fils de liaison.

Vous pouvez désactiver le câblage automatique de tous les nouveaux fils de liaison en sélectionnant **Outils»Options** et en sélectionnant **Diagramme** dans le menu déroulant du haut de la fenêtre.

Si vous désactivez le câblage automatique, 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.

Vous pouvez aussi appuyer sur la barre espace pour alterner entre le sens vertical et horizontal si le câblage automatique est activé. Si LabVIEW trouve une trajectoire pour le fil de liaison dans la nouvelle direction, celui-ci utilise la nouvelle direction.

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.

Correction des fils de liaison brisés

Un fil de liaison brisé apparaît sous la forme d'un trait noir en pointillés avec un symbole x rouge au milieu. 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 sur un fil de liaison brisé pour visualiser l'info-bulle qui décrit le motif de rupture du fil de liaison. Cette information apparaît aussi dans la fenêtre d'**aide contextuelle** lorsque vous placez l'outil Bobine sur un fil de liaison brisé. Cliquez avec le bouton droit sur le fil de liaison et

sélectionnez **Liste des erreurs** dans le menu local pour afficher la boîte de dialogue **Liste des erreurs**. Cliquez sur le bouton **Aide** pour obtenir plus d'informations sur la cause du fil de liaison 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. Vous pouvez aussi cliquer avec le bouton droit sur le fil de liaison et sélectionner dans le menu local des options telles que **Supprimer une branche de câblage**, **Créer une branche de câblage**, **Supprimer les segments déconnectés**, **Arranger le câblage**, **Changer en commande**, **Changer en indicateur**, **Activer l'indexation à la source** et **Désactiver l'indexation à la source**. Ces options sont disponibles ou non en fonction de l'origine du fil de liaison brisé.

Pour supprimer tous les fils de liaison brisés, sélectionnez **Édition» Supprimer les fils brisés** ou appuyez simultanément sur les touches <Ctrl-B>. (**Mac OS**) Appuyez sur les touches <Command-B>. (**UNIX**) Appuyez sur les touches <Meta-B>.



Mise en garde Soyez prudent quand vous supprimez tous les fils de liaison brisés. Parfois un fil de liaison semble être brisé car vous n'avez pas terminé le câblage du diagramme.

Points de coercition

Les points de coercition apparaissent sur les nœuds du diagramme pour vous avertir que vous avez câblé ensemble deux types de données différents. Le point signifie que LabVIEW a converti la valeur transmise au nœud en une représentation différente. Par exemple, la fonction Additionner attend deux entrées à virgule flottante double précision. Si vous remplacez l'une de ces entrées par un entier, un point de coercition apparaît sur la fonction Additionner.

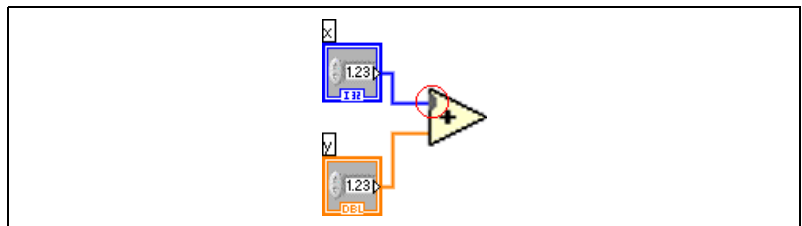


Figure 5-2. Point de coercition

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.

Des points de coercition apparaissent également sur le terminal quand vous câblez des données à un terminal variant, sauf si vous câblez ensemble deux données de type variant. Reportez-vous à la section *Manipulation de données variant* de ce chapitre pour obtenir de plus amples informations sur le type de données variant.

Les points de coercition peuvent provoquer une utilisation de mémoire plus importante de la part d'un VI et augmenter son temps 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 d'entrée ou de sortie unique. Un VI polymorphe correspond à un ensemble de sous-VIs avec les mêmes modèles de connecteur. Chaque sous-VI correspond à une instance de VI polymorphe.

Par exemple, le VI Lire une clé est polymorphe. Son terminal **valeur par défaut** accepte des booléens, des nombres à virgule flottante double précision, des nombres entiers signés 32 bits, des chemins, des chaînes ou des nombres entiers non signés 32 bits.

Pour la plupart des VIs polymorphes, le type de données que vous câblez aux entrées du VI polymorphe détermine l'instance à utiliser. Si un VI polymorphe ne contient pas de sous-VI pour ce type de données, un fil de liaison brisé apparaît. Si le type de données que vous câblez au VI polymorphe ne détermine pas l'instance à utiliser, vous devez sélectionner l'instance manuellement. Si vous sélectionnez l'instance d'un VI polymorphe manuellement, le VI ne se comporte plus comme un VI polymorphe car il n'accepte et ne renvoie plus que les types de données de l'instance que vous avez sélectionnée.

Pour sélectionner l'instance manuellement, cliquez avec le bouton droit sur le VI polymorphe, sélectionnez **Sélectionner le type** dans le menu local et

Nom du VI d'instance ▾

sélectionnez ensuite l'instance à utiliser. Vous pouvez aussi utiliser l'outil Doigt pour cliquer sur le sélecteur du VI polymorphe, représenté à gauche, et sélectionner une instance dans le menu local. Cliquez avec le bouton droit sur le VI polymorphe du diagramme et sélectionnez **Éléments visibles**»**Sélecteur du VI polymorphe** dans le menu local pour afficher le sélecteur. Pour permettre à un VI polymorphe d'accepter à nouveau tous les types de données gérés, cliquez avec le bouton droit sur le VI polymorphe et sélectionnez **Sélectionner le type**»**Automatique** dans le menu local, ou cliquez sur le sélecteur du VI polymorphe avec l'outil Doigt et sélectionnez **Automatique** dans le menu local.

Construction de VIs polymorphes

Construisez vos propres VIs polymorphes lorsque vous effectuez la même opération sur différents types de données.



Remarque Vous ne pouvez construire et modifier des VIs polymorphes que dans le système de développement professionnel 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 5-3.

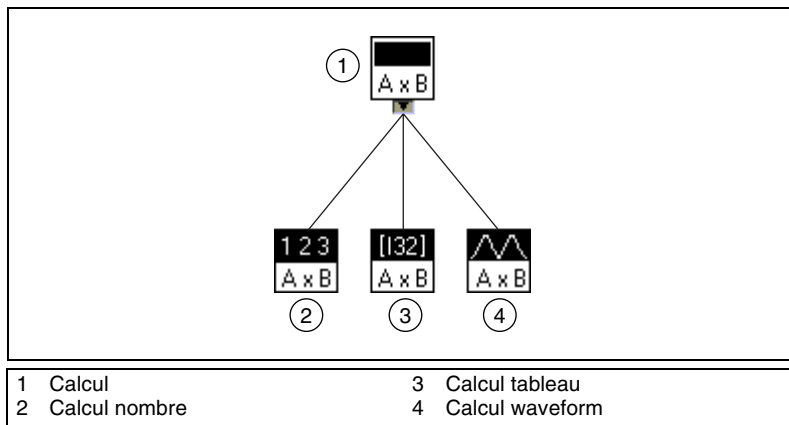


Figure 5-3. Exemple d'un VI polymorphe

Le VI Calcul appelle de manière statique 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 5-4.

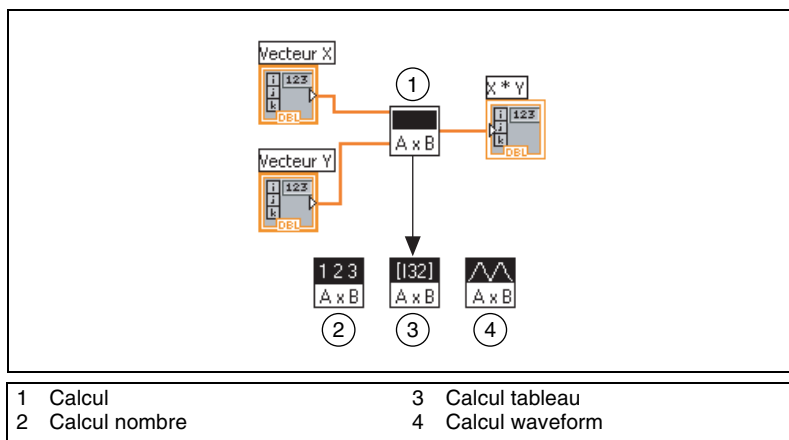


Figure 5-4. VI polymorphe appelant un sous-VI

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 section de la documentation portant sur 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 sur les fonctions polymorphes.

VIs Express

Utilisez les VIs Express pour effectuer des tâches de mesure courantes. Les VIs Express sont des nœuds de fonction qui ne nécessitent qu'un câblage minimal car vous les configurez à l'aide des boîtes de dialogue. Les entrées et sorties d'un VI Express dépendent de la manière dont vous avez configuré ce VI. Les VIs Express apparaissent sur le diagramme sous forme de nœuds extensibles avec des icônes entourées d'un champ bleu.

Reportez-vous à la section *Affichage de sous-VIs et de VIs Express sous forme d'icônes ou de nœuds extensibles* du chapitre 7, *Création de VIs et de sous-VIs*, pour obtenir de plus amples informations sur les nœuds extensibles. Reportez-vous au manuel *Initiation à LabVIEW* pour obtenir des informations complémentaires sur les VIs Express.

Création de VIs Express en sous-VIs

Vous pouvez créer un sous-VI à partir de la configuration d'un VI Express. Par exemple, enregistrez la configuration du VI Express *Écrire un fichier de mesures LabVIEW* pour pouvoir l'utiliser comme sous-VI dans d'autres VIs que vous créez plutôt que d'avoir à reconfigurer chaque fois le VI Express. Cliquez avec le bouton droit sur le VI Express et sélectionnez **Ouvrir la face-avant** dans le menu local pour créer un sous-VI à partir d'un VI Express.

Lorsque vous créez un sous-VI à partir d'un VI Express, la face-avant du sous-VI s'affiche. Vous pouvez alors modifier le VI et l'enregistrer. Pour enregistrer les valeurs entrées dans les commandes, sélectionnez **Exécution» Désigner les valeurs actuelles comme défaut** ou cliquez avec le bouton droit sur chaque commande et sélectionnez **Opérations sur les données» Désigner la valeur actuelle comme défaut** dans le menu local. Un nouveau sous-VI affiché comme nœud extensible remplace le VI Express sur le diagramme.

Une fois que vous avez créé un VI à partir d'un VI Express, vous ne pouvez pas reconvertir ce sous-VI en VI Express.

Type de données dynamiques



Le type de données dynamiques apparaît sous la forme d'un terminal bleu foncé, représenté à gauche. La plupart des VIs Express acceptent et/ou renvoient le type de données dynamiques. Vous pouvez câbler le type de données dynamiques à n'importe quel indicateur ou entrée qui accepte les données numériques, waveform ou booléennes. Câblez le type de données

dynamiques à l'indicateur capable de présenter les données de la meilleure manière. Les indicateurs comprennent un graphe, un graphe déroulant ou un indicateur numérique.

Le type de données dynamiques est prévu pour être utilisé avec les VIs Express. La plupart des autres VIs et fonctions livrés avec LabVIEW n'acceptent pas ce type de données. Pour utiliser un VI ou une fonction intégré en vue d'analyser ou de traiter les données comprises dans le type de données dynamiques, vous devez convertir le type de données dynamiques. Reportez-vous à la section [Convertir des données dynamiques](#) de ce chapitre pour obtenir de plus amples informations sur la conversion du type de données dynamiques.

Outre les données associées au signal, le type de données dynamiques inclut des attributs qui fournissent des informations sur le signal, comme le nom du signal ou l'heure et la date auxquelles les données ont été acquises. Les attributs spécifient de quelle manière le signal est représenté sur un graphe ou un graphe déroulant. Par exemple, si vous utilisez le VI Express Assistant DAQ pour acquérir un signal et que vous tracez ce signal sur un graphe, le nom du signal apparaît dans la légende du graphe et l'échelle x s'ajuste pour afficher les informations de synchronisation associées au signal en temps relatif ou absolu selon les attributs du signal. Si vous utilisez le VI Express Mesures spectrales pour effectuer une analyse FFT sur le signal et que vous représentez la valeur résultante sur un graphe, l'échelle x s'ajuste automatiquement pour représenter le signal dans le domaine fréquentiel en fonction des attributs de ce signal. Effectuez un clic droit sur un terminal de sortie de type de données dynamiques d'un VI ou d'une fonction sur le diagramme et sélectionnez **Créer»Indicateur graphe** pour afficher les données dans un graphe ou sélectionnez **Créer»Indicateur numérique** pour afficher les données dans un indicateur numérique.

Table 5-2 dresse la liste des indicateurs qui acceptent le type de données dynamiques et le type de données que le type de données dynamiques est susceptible de contenir, et décrit la manière dont les indicateurs gèrent les données.

Tableau 5-2. Indicateurs de type de données dynamiques

Données contenues dans le type de données dynamiques	Indicateur	Résultat
Valeur numérique unique	Graphe	Trace la valeur unique, y compris l'horodatage et les attributs
Voie unique		Trace la waveform complète, y compris l'horodatage et les attributs
Voies multiples		Trace toutes les données, y compris leur horodatage et leurs attributs
Valeur numérique unique	Indicateur numérique	Affiche la valeur unique
Voie unique		Affiche la dernière valeur des données de la voie
Voies multiples		Affiche la dernière valeur des données de la première voie
Valeur numérique unique	Indicateur booléen	Affiche la valeur VRAI si la valeur numérique est supérieure ou égale à 0,5
Voie unique		Affiche la valeur VRAI si la dernière valeur des données de la voie est supérieure ou égale à 0,5
Voies multiples		Affiche la valeur VRAI si la dernière valeur des données de la première voie est supérieure ou égale à 0,5

Convertir des données dynamiques

Utilisez le VI Express Convertir des données dynamiques pour convertir le type de données dynamiques en types de données numériques, waveform et tableau afin de les utiliser avec d'autres VIs et fonctions. Lorsque vous placez le VI Express Convertir des données dynamiques sur le diagramme, la boîte de dialogue **Configurer convertir des données dynamiques** s'affiche. La boîte de dialogue **Configurer convertir des données dynamiques** affiche des options vous permettant de spécifier de quelle manière vous voulez formater les données renvoyées par le VI Express Convertir des données dynamiques.

Par exemple, si vous acquérez un signal sinusoïdal à partir d'un périphérique d'acquisition de données, sélectionnez l'option **Waveform simple** sur la boîte de dialogue **Configurer convertir des données dynamiques**. Câblez la sortie **Waveform** du VI Express Convertir des

données dynamiques à une fonction ou à un VI qui accepte le type de données waveform. Si vous avez acquis un certain nombre de températures à partir de différentes voies à l'aide d'un périphérique DAQ, sélectionnez les options **Valeurs les plus récentes de chaque voie** et **Nombres à virgule flottante (double)**. Câblez ensuite la sortie **Tableau** du VI Express Convertir des données dynamiques à une fonction ou à un VI qui accepte un tableau numérique comme entrée.

Lorsque vous câblez un type de données numériques à un indicateur de tableau, LabVIEW place automatiquement le VI Express Convertir des données dynamiques sur le diagramme. Double-cliquez sur le VI Express Convertir des données dynamiques pour ouvrir la boîte de dialogue **Configurer convertir des données dynamiques** pour définir l'affichage des données dans le tableau.

Conversion en données dynamiques

Utilisez le VI Express Convertir en données dynamiques pour convertir des types de données numériques, booléens, waveform et tableau en données dynamiques pour pouvoir les utiliser avec les VIs Express. Lorsque vous placez le VI Express Convertir en données dynamiques sur le diagramme, la boîte de dialogue **Configurer convertir en données dynamiques** s'affiche. Utilisez cette boîte de dialogue pour sélectionner le type de données à convertir en type de données dynamiques.

Par exemple, si vous acquérez un signal sinusoïdal à l'aide des VIs NI-DAQ traditionnel d'entrée analogique et que vous voulez utiliser un VI Express Analyse de signal pour analyser le signal, sélectionnez l'option Waveform simple dans la boîte de dialogue **Configurer convertir en données dynamiques**. Connectez ensuite la sortie **Type de données dynamiques** à un VI Express qui accepte le type de données dynamiques comme entrée.

Manipulation de données variant

Les données variant ne se conforment pas à un type de données spécifique et peuvent contenir des attributs. LabVIEW représente les données variant avec le type de données variant. Le type de données variant se distingue des autres types de données car il enregistre le nom de l'indicateur ou de la commande, des informations sur le type de données à partir desquelles vous avez effectué la conversion et les données elles-mêmes, ce qui permet à LabVIEW de convertir correctement le type de données variant en données du type de votre choix. 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 les fonctions Variant pour créer et manipuler des données 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. Plusieurs fonctions polymorphes renvoient le type de données variant.

Utilisez le type de données variant lorsqu'il est essentiel de manipuler les données indépendamment de leur type, comme par exemple quand vous transmettez ou que vous stockez des données, que vous lisez et/ou écrivez à partir ou vers des dispositifs inconnus ou que vous stockez des informations dans une pile, une file d'attente ou un notificateur, ou encore que vous effectuez des opérations sur un jeu de commandes hétérogènes.

Vous pouvez aussi utiliser la fonction Aplatir en chaîne pour convertir un type de données en données de type chaînes afin de représenter des données indépendamment de leur type. Aplatir les données en chaînes s'avère utile lorsque vous utilisez TCP/IP pour transmettre des données car ce protocole ne sait interpréter que les chaînes.

Toutefois, l'utilisation de données aplaties n'est pas sans limites, puisque LabVIEW ne peut contraindre les données aplaties quand le type de données d'origine ne correspond pas au type de données dans lequel vous voulez réaliser la conversion. De plus, la tentative de redresser un entier aplati sous la forme d'un nombre à virgule flottante de précision étendue se solde par un échec. Reportez-vous à la section *Flattened Data* de la note d'application *LabVIEW Data Storage* pour obtenir de plus amples informations sur l'aplatissement et le redressement des données.

Un autre avantage de l'utilisation du type de données variant est que ce type est capable de stocker les attributs des données. Un attribut est un ensemble d'informations sur les données que le type de données variant enregistre. Par exemple, si vous voulez connaître l'heure à laquelle des données ont été créées, enregistrez-les comme données variant et ajoutez-leur un attribut nommé **Heure** permettant de stocker la chaîne d'heure. Les données des attributs peuvent être de n'importe quel type. Utilisez des attributs variant quand vous voulez trier les données en fonction d'un attribut particulier, identifier le périphérique ou l'application qui a généré les données ou filtrer les données pour trouver uniquement les variants comportant un attribut particulier.

Unités numériques et vérification de type stricte

Vous pouvez associer des unités physiques de mesure, comme des mètres ou des kilomètres par seconde, à n'importe quelle commande ou indicateur 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 effectuant un clic droit sur la commande et en sélectionnant **Éléments visibles** » **Étiquette d'unité** dans le menu local. Cliquez avec le bouton droit sur une étiquette d'unité et sélectionnez **Construire la chaîne d'unité** dans le menu local pour modifier l'étiquette d'unité.

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.

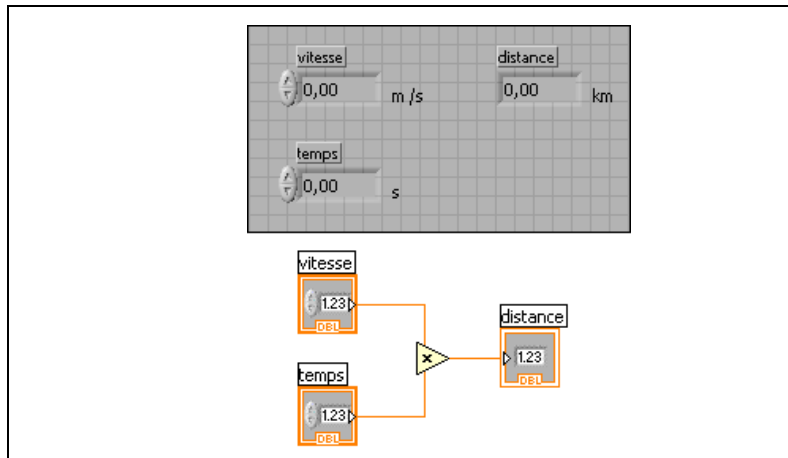


Figure 5-5. Câblage d'objets avec des unités compatibles

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.

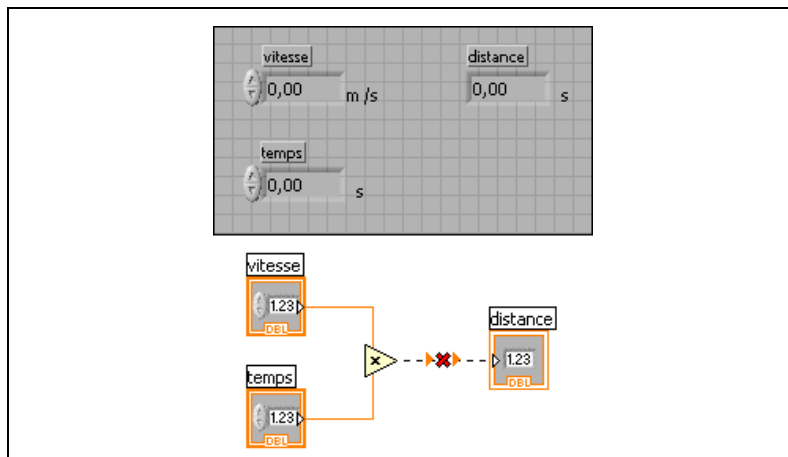


Figure 5-6. La connexion de sources avec des unités incompatibles se traduit par des fils de liaison brisés

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'incrémentatıon est ambiguë en ce qui concerne les unités. Si vous utilisez des unités de distance, la fonction d'incrémentatıon 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'incrémement, 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.

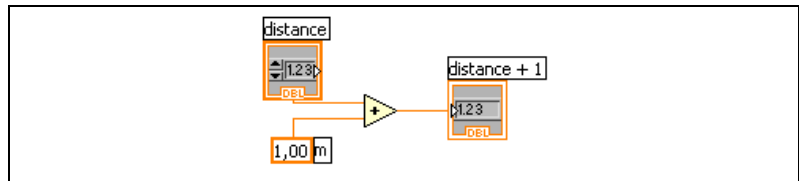


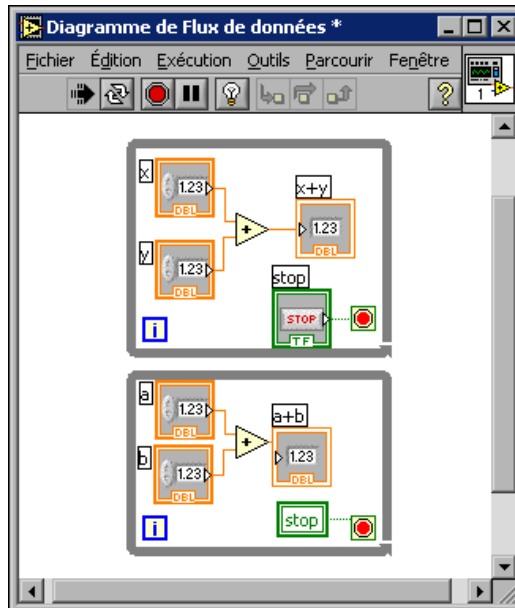
Figure 5-7. Création d'une fonction d'incrémement avec des unités

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. Lorsque qu'un nœud termine une exécution, il fournit les données à ses terminaux de sortie et transmet les données de sortie 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 que l'ordre séquentiel des commandes pour déterminer l'ordre d'exécution des éléments du diagramme, vous pouvez créer des diagrammes ayant des opérations simultanées. Vous pouvez par exemple exécuter deux boucles While simultanément et afficher les résultats sur la face-avant.



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 artificielle des données

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 naturelle des données. Reportez-vous à la section *Structures Séquence* du chapitre 8, *Boucles et structures*, pour obtenir de plus amples informations sur les structures Séquence. Vous

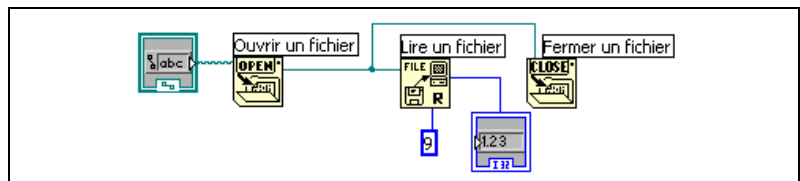
pouvez également utiliser des paramètres dupliqués pour contrôler l'ordre d'exécution. Reportez-vous à la section *Paramètres dupliqués* du chapitre 14, *E/S sur fichiers*, pour de plus amples informations sur les paramètres dupliqués.

Une autre façon de contrôler l'ordre d'exécution est de créer une dépendance artificielle des données, dans laquelle le nœud récepteur n'utilise pas réellement 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 `exemples\general\structs.llb` pour étudier un exemple de dépendance artificielle des données.

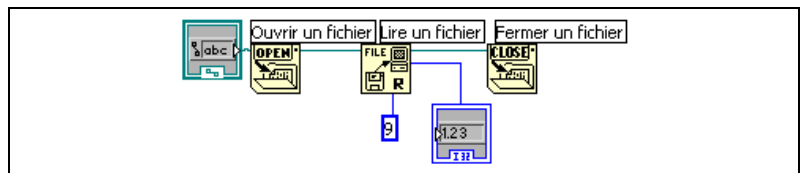
Dépendance de données manquante

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 de mémoire. Reportez-vous à la section *Memory and Speed Optimization* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur l'optimisation de l'utilisation de la mémoire lors de votre travail de développement.

Conception du diagramme

Utilisez les conseils suivants pour concevoir vos diagrammes :

- Utilisez une présentation 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 obtenir 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 *Détection 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 du câblage ne produit pas forcément d'erreurs, mais peut compliquer la lecture ou la mise au point du diagramme ou faire croire à tort que le VI effectue certaines opérations.
- 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 placer des objets sur les fils de liaison. Les fils de liaison ne se connectent qu'aux objets sur lesquels vous cliquez. Le fait de placer un terminal ou une icône sur un fil de liaison peut faire croire à tort qu'une connexion est présente à cet endroit-là.
- Utilisez des étiquettes libres pour documenter du code sur le diagramme.
- Augmentez l'espace entre des objets entassés ou groupés de 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.

(Mac OS) 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 combinaison de touches pour ajouter l'espace.

Reportez-vous à la section *Block Diagram Style* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la conception du 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 de plus amples informations sur l'exécution et la mise au point des VIs.

Exécution des VIs




L'exécution d'un VI lance 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 sous-VI si vous créez un connecteur pour le VI. Tant que le VI est en cours d'exécution, le bouton **Exécuter** se transforme en flèche assombrie, représentée à gauche, pour indiquer que le VI est en train de s'exécuter. Vous ne pouvez pas modifier un VI pendant son exécution.



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**, représenté à gauche, exécute le VI en continu jusqu'à ce que vous l'arrêtiez manuellement. 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 obtenir de plus amples 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 arrêter 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 servant à arrêter le VI.

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 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 des informations complémentaires sur la création de VIs multithreads. Consultez le chapitre 16, *Personnalisation des VIs*, pour plus de précisions sur la manière de configurer l'exécution des VIs.



Remarque La configuration incorrecte d'un VI pour qu'il s'exécute selon différentes priorités risque de provoquer une exécution inattendue du VI. Vous n'avez pas besoin de configurer la plupart des VIs pour qu'ils s'exécutent selon différentes priorités.

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 à gauche, quand le VI que vous créez ou que vous modifiez contient des erreurs. S'il apparaît toujours brisé après que vous avez terminé le câblage du diagramme, le VI est brisé et ne pourra pas s'exécuter.

Recherche des causes de 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 **Afficher l'erreur** ou double-cliquez sur la description de l'erreur pour afficher le diagramme ou la face-avant concerné et mettre en surbrillance l'objet contenant l'erreur.



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

Configurez LabVIEW pour qu'il affiche systématiquement 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 la case **Afficher les mises en garde dans la liste des erreurs par défaut**. 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 conçues pour vous aider à éviter des problèmes éventuels dans les VIs.

Causes fréquentes 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 obtenir de plus amples informations sur le câblage des objets dans le diagramme.
- Il manque un câblage nécessaire à un terminal du diagramme. Reportez-vous à la section *Définition des entrées et sorties nécessaires, recommandées et facultatives* du chapitre 7, *Création de VIs et de sous-VIs*, pour obtenir de plus amples informations sur les terminaux requis.
- 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 obtenir de plus amples informations sur les sous-VIs.

Techniques de mise au point

Si un VI n'est pas brisé, mais que vous obtenez des données inattendues, vous pouvez utiliser les techniques suivantes pour identifier et corriger les problèmes d'un VI ou du flux de données du 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 obtenir de plus amples informations sur l'utilisation de ces paramètres.
- Pour éliminer toutes les mises en garde, sélectionnez **Fenêtre» Afficher la liste des erreurs** et cochez la case **Afficher les mises en garde** pour voir toutes les mises en garde concernant le VI. Déterminez les causes et éliminez-les du VI.
- Cliquez trois fois sur le fil de liaison avec l'outil Flèche pour sélectionner l'ensemble de son chemin et pour vous assurer que les fils de liaison se connectent aux terminaux appropriés.
- Utilisez la fenêtre d'**Aide contextuelle** pour vérifier les valeurs par défaut de chaque fonction et sous-VI du diagramme. Les VIs et fonctions transmettent des valeurs par défaut si des entrées optionnelles ou nécessaires ne sont pas câblées. Par exemple, une entrée booléenne pourrait être définie à VRAI si elle n'est pas câblée.
- Utilisez la boîte de dialogue **Rechercher** pour rechercher des sous-VIs, du texte et d'autres objets afin de corriger votre application.
- Sélectionnez **Parcourir»Hiérarchie du VI** pour rechercher les sous-VIs non câblés. À l'inverse des fonctions non câblées, les VIs non câblés ne génèrent pas toujours des erreurs, sauf si vous configurez une entrée comme entrée obligatoire. Si vous placez par erreur un sous-VI non câblé sur le diagramme, il s'exécute en même temps que le diagramme. Par conséquent, votre application risque d'effectuer des actions supplémentaires.
- 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 observer des valeurs de données intermédiaires et pour vérifier la sortie d'erreur des VIs et fonctions, en particulier ceux qui réalisent des opérations d'E/S.

- 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.
- Déterminez si les données transmises par une fonction ou un sous-VI ne sont pas définies. Cela se produit fréquemment avec des nombres. Par exemple, à un certain moment de votre VI, une opération peut avoir divisé un nombre par zéro, renvoyant ainsi **Inf** (infini), alors que les fonctions ou les sous-VIs ultérieurs attendaient des nombres.
- Si votre VI s'exécute plus lentement que prévu, assurez-vous que vous avez désactivé l'animation de l'exécution dans les sous-VIs. De même, fermez les diagrammes et les faces-avant des sous-VIs lorsque vous ne les utilisez pas car les fenêtres ouvertes peuvent affecter la vitesse d'exécution.
- Vérifiez la représentation des commandes et indicateurs pour voir si vous recevez un dépassement de capacité parce que vous avez converti un nombre à virgule flottante en un entier ou un entier en un entier plus petit. Par exemple, il arrive que vous câbliez un entier 16 bits à une fonction qui n'accepte que les entiers 8 bits. En effet, la fonction convertit l'entier 16 bits en une représentation 8 bits, provoquant éventuellement une perte de données.
- Déterminez si des boucles For n'exécutent pas par erreur zéro itération et produisent des tableaux vides.
- Vérifiez que vous avez correctement initialisé les registres à décalage, à moins que vous ne souhaitiez qu'ils enregistrent des données d'une exécution de la boucle à une autre.
- Vérifiez l'ordre des éléments des clusters aux points source et destination. LabVIEW détecte la non-correspondance du type de données et de la taille du cluster lors de l'édition, mais il ne détecte pas les non-correspondances d'éléments du même type.
- Vérifiez l'ordre d'exécution du nœud.
- Vérifiez que le VI ne contient pas de sous-VI masqués. Vous pourriez avoir masqué par inadvertance un sous-VI en le déposant directement sur un autre nœud, en diminuant la taille d'une structure sans conserver le sous-VI dans la vue.
- De même, vérifiez l'inventaire des sous-VIs que le VI utilise et comparez les résultats de **Parcourir»Sous-VIs appelés par ce VI** et

Parcourir»Sous-VIs non ouverts afin de déterminer si des sous-VIs supplémentaires existent. Vous pouvez aussi ouvrir la fenêtre **Hiérarchie** pour voir les sous-VIs d'un VI. Pour éviter d'obtenir des résultats incorrects provoqués par des VIs masqués, spécifiez que les entrées aux VIs sont nécessaires.

Mode Animation



Affichez l'animation de l'exécution du diagramme en cliquant sur le bouton **Mode Animation**, représenté à 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 entre les nœuds d'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 obtenir des informations complémentaires sur les clusters d'erreur.

Mode pas à pas



Avancez pas à pas dans un VI pour voir chaque action du VI sur le diagramme lors de son 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 au-dessus du 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, représenté à gauche, apparaît sur les icônes des sous-VIs en cours d'exécution.

Outil Sonde



Utilisez l’outil Sonde, représenté à 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 renvoyer 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. Lors de la suspension de l’exécution à un nœud en raison du mode pas à pas ou d’un point d’arrêt, vous pouvez aussi sonder le fil de liaison qui vient de s’exécuter pour connaître la valeur qui a été transmise par ce fil de liaison.

Types de sondes

Vous pouvez vérifier les valeurs intermédiaires sur un fil de liaison durant l’exécution d’un VI au moyen de la Sonde générique, en utilisant un indicateur sur la palette de **Commandes** pour afficher les données, en utilisant une sonde fournie, une sonde personnalisée fournie ou en créant une nouvelle sonde.

Générique

Utilisez une sonde générique pour afficher les données qui sont transmises par un fil de liaison. Cliquez avec le bouton droit de la souris sur un fil de liaison et sélectionnez **Sonde personnalisée»Générique** dans le menu local pour utiliser la sonde générique.

La sonde générique affiche les données. Vous ne pouvez pas configurer la sonde générique pour répondre aux données.

LabVIEW affiche la sonde générique lorsque vous cliquez avec le bouton droit sur un fil de liaison et que vous sélectionnez **Sonde**, sauf si vous avez déjà spécifié une sonde fournie ou une sonde personnalisée pour ce type de données.

Vous pouvez mettre au point une sonde personnalisée comme s’il s’agissait d’un VI. Toutefois, une sonde ne peut sonder son propre diagramme, ni celui de ses sous-VIs. Quand vous mettez au point des sondes, utilisez la sonde générique.

Utilisation d'indicateurs pour afficher les données

Vous pouvez aussi utiliser un indicateur pour afficher les données transmises par un fil de liaison. Par exemple, si vous affichez des données numériques, vous pouvez utiliser un graphe déroulant à l'intérieur de la sonde pour afficher les données. Cliquez avec le bouton droit sur un fil de liaison, sélectionnez **Sonde personnalisée»Commandes** dans le menu local et sélectionnez l'indicateur que vous voulez utiliser. Vous pouvez aussi cliquer sur l'icône **Sélectionner une commande** dans la palette de **Commandes** et sélectionner une commande ou une définition de type enregistrée dans l'ordinateur ou dans un répertoire partagé sur un serveur. LabVIEW traite les définitions de type comme des commandes personnalisées lorsque vous les utilisez pour afficher des données sondées.

Si le type de données de l'indicateur que vous avez sélectionné ne correspond pas au type de données du fil de liaison sur lequel vous avez effectué un clic droit, LabVIEW ne place pas l'indicateur sur le fil de liaison.

Sondes fournies

Les sondes fournies sont des VIs qui affichent des informations complètes sur les données transmises par un fil de liaison. Par exemple, le VI Sonde Refnum renvoie des informations sur le nom du VI, le chemin du VI et la valeur hexadécimale de la référence. Vous pouvez aussi utiliser une sonde fournie pour répondre en fonction des données transmises à travers le fil de liaison. Par exemple, utilisez une sonde Erreur ou un cluster d'erreur pour recevoir l'état, le code, la source et la description de l'erreur et spécifier si vous voulez définir un point d'arrêt conditionnel si une erreur ou une mise en garde a lieu.

Les sondes fournies apparaissent en haut du menu local **Sonde personnalisée**. Cliquez avec le bouton droit de la souris sur un fil de liaison et sélectionnez **Sonde personnalisée** dans le menu local pour sélectionner une sonde fournie. Seules les sondes qui correspondent au type de données du fil de liaison sur lequel vous avez effectué un clic droit apparaissent dans le menu local.

Reportez-vous au VI Using Supplied Probes VI dans `examples\general\probes.llb` pour obtenir un exemple d'utilisation des sondes fournies.

Sondes personnalisées

Utilisez l'Assistant des sondes personnalisées pour créer une sonde basée sur une sonde existante ou pour en créer une nouvelle. Cliquez avec le bouton droit sur un fil de liaison et sélectionnez **Sonde personnalisée» Nouveau** dans le menu local pour afficher l'Assistant des sondes personnalisées. Créez une sonde lorsque vous voulez exercer un contrôle plus étroit sur la manière dont LabVIEW sonde les données transmises par l'intermédiaire d'un fil de liaison. Lorsque vous créez une nouvelle sonde, le type de données de la sonde correspond au type de données du fil de liaison sur lequel vous avez effectué un clic droit. Si vous voulez modifier la sonde que vous avez créée, vous devez l'ouvrir à partir du répertoire dans lequel vous l'avez enregistrée.

Une fois que vous avez sélectionné une sonde dans le menu **Sonde personnalisée**, naviguez jusqu'à celle-ci à l'aide de l'option de palette **Sélectionner une commande** ou créez une nouvelle sonde au moyen de l'Assistant des sondes personnalisées ; cette sonde devient alors la sonde par défaut pour ce type de données et celle que LabVIEW charge lorsque vous effectuez un clic droit sur un fil de liaison et que vous sélectionnez **Sonde** dans le menu local. LabVIEW ne charge que les sondes qui correspondent exactement au type de données du fil de liaison sur lequel vous avez effectué un clic droit. C'est ainsi qu'une sonde numérique à virgule flottante double précision ne peut pas sonder un fil de liaison entier 32 bits non signé même si LabVIEW est capable de convertir les données.



Remarque Si vous voulez qu'une sonde personnalisée soit la sonde par défaut pour un type de données particulier, enregistrez cette sonde dans le répertoire `user.lib_probes\default`. N'enregistrez pas les sondes dans le répertoire `vi.lib_probes` car LabVIEW écrase ces fichiers lorsque vous effectuez une mise à jour ou une réinstallation.

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations complémentaires sur les mises en garde au cours de l'utilisation et de la création des sondes personnalisées.

Points d'arrêt



Utilisez l'outil Point d'arrêt, représenté à 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 se met en pause

après le passage des données dans le fil de liaison. Placez un point d'arrêt sur le diagramme pour suspendre l'exécution une fois que tous les nœuds du diagramme ont été exécutés.

Quand un VI effectue une pause à un point d'arrêt, LabVIEW amène le diagramme à l'avant de la fenêtre et fait clignoter le nœud ou le fil de liaison qui contient le point d'arrêt. Lorsque vous placez le curseur sur un point d'arrêt existant, la zone noire de l'outil Point d'arrêt devient blanche.

Lorsque l'exécution atteint un point d'arrêt, le VI se met en pause et le bouton **Pause** est représenté en rouge. Vous pouvez effectuer les 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.
- Modifier les valeurs des commandes de la face-avant.
- Cliquer sur le bouton **Pause** pour poursuivre l'exécution jusqu'au prochain point d'arrêt ou jusqu'à la fin du VI.

LabVIEW enregistre les points d'arrêt avec un VI, mais ceux-ci ne sont activés que lors de l'exécution. Vous pouvez afficher tous les points d'arrêt en sélectionnant **Parcourir»Points d'arrêt**.

Suspension de l'exécution

Suspendez l'exécution d'un sous-VI pour modifier les valeurs des commandes et indicateurs, pour contrôler le nombre d'exécutions avant de retourner à l'appelant ou pour revenir au début de l'exécution d'un 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 est automatiquement suspendu 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 est suspendu 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 la case **Interrompre l'exécution à l'appel** pour suspendre l'exécution uniquement à cette instance du sous-VI.

La fenêtre **H hiérarchie**, que vous affichez en sélectionnant **Parcourir» Hiérarchie du VI**, indique si un VI est en pause ou suspendu. 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 **Liste d'appel** sur la barre d'outils liste la chaîne des appelants du VI de niveau principal au sous-VI. Il ne s'agit pas de la même liste que celle 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 **Liste d'appel**, son diagramme s'ouvre et LabVIEW met en surbrillance l'instance actuelle du sous-VI.

Mise "en commentaire" de sections particulières du diagramme

Vous pouvez exécuter un VI en désactivant une partie du diagramme, de la même manière que l'on transforme en commentaire une section de code dans un langage de programmation textuel. Désactivez une section du diagramme pour déterminer si le VI fonctionne mieux sans cette dernière.

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 et structures](#), pour obtenir de plus amples 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 afin de réduire les exigences de mémoire et d'améliorer légèrement les performances. Cliquez avec le bouton droit sur le connecteur et sélectionnez **Propriétés du VI**. Sélectionnez **Exécution** dans le menu déroulant **Catégorie** et désélectionnez l'option **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- renvoient 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 à virgule flottante propagent NaN et Inf de manière fiable. Lorsque vous convertissez explicitement ou implicitement des NaN ou Inf en entiers ou en valeurs booléennes, les valeurs deviennent incohérentes. Par exemple, la division de 1 par zéro produit Inf. La conversion de Inf en un entier 16 bits produit la valeur 32 767, 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 dans des boucles

Les boucles For renvoient des données par défaut lorsque le décompte d'itération est égal à zéro.

Reportez-vous à la section *Type de données des indicateurs et des commandes* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur les valeurs par défaut pour les types de données.

Boucles For

Les boucles For produisent des valeurs par défaut si vous câblez 0 au terminal de décompte de la boucle For ou si vous câblez un tableau vide comme entrée à la boucle For alors que l'auto-indexation est activée. La boucle ne s'exécute pas, et toutes les sorties de tunnels pour lesquelles l'auto-indexation est désactivée contiennent la valeur par défaut pour le type de données du tunnel. 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 boucles For et boucles While* du chapitre 8, *Boucles et structures*, pour obtenir de plus amples informations sur les boucles For, l'auto-indexation et les registres à décalage.

Données par défaut dans les tableaux

L'indexation au-delà des limites d'un tableau produit la valeur par défaut pour le paramètre 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 10, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur les tableaux. Reportez-vous à la section *Auto-indexation des boucles* du chapitre 8, *Boucles et structures*, pour obtenir de plus amples 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 aux valeurs spéciales telles que NaN, Inf ou des tableaux vides pour déterminer si un VI produit des données non définies. 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 comment vous voulez que le VI réagisse lorsque le tableau d'entrée est vide : Il peut soit produire un code d'erreur de sortie, soit remplacer les données définies par des valeurs créées par la boucle.

Détection et gestion d'erreur

Chaque erreur possède un code numérique et un message d'erreur correspondant. Par défaut, LabVIEW gère automatiquement toute erreur lors de l'exécution du VI en interrompant son exécution, en mettant en évidence le sous-VI ou la fonction où l'erreur s'est produite et en affichant la boîte de dialogue **Erreur**.

Sélectionnez **Fichier»Propriétés du VI** et sélectionnez **Exécution** dans le menu déroulant **Catégorie** pour désactiver la gestion d'erreur automatique pour un VI spécifique. Sélectionnez **Outils»Options** et sélectionnez ensuite **Diagramme** dans le menu déroulant du haut de la fenêtre afin de désactiver la gestion d'erreur automatique pour tous les nouveaux VIs que vous créez.

Si vous voulez qu'un sous-VI ou une fonction ignore toute erreur rencontrée, câblez la sortie **sortie d'erreur** du sous-VI ou de la fonction à l'entrée **entrée d'erreur** du VI Effacer les erreurs. Pour désactiver la gestion d'erreur automatique pour un sous-VI ou une fonction, câblez son paramètre **sortie d'erreur** au paramètre **entrée d'erreur** d'un autre sous-VI ou d'une autre fonction, ou à un indicateur **sortie d'erreur**.

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 d'utiliser la gestion d'erreur.

Détection d'erreurs

Même si vous êtes sûr des VIs que vous créez, vous ne pouvez pas prédire les problèmes que rencontreront les utilisateurs. Sans mécanisme de recherche d'erreur, vous savez seulement que le VI ne fonctionne pas correctement. La détection d'erreurs vous indique pour quelle raison et à quel endroit l'erreur s'est produite.

Lorsque vous effectuez des opérations d'entrées/sorties (E/S), envisagez l'éventualité d'erreurs. Presque toutes les fonctions d'E/S renvoient 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 les 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

Par défaut, LabVIEW gère automatiquement les erreurs en interrompant l'exécution. Vous pouvez choisir d'autres méthodes de gestion des erreurs. 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 obtenir des informations complémentaires sur les clusters d'erreur.

La gestion d'erreur dans LabVIEW suit le modèle de flux de données. Les informations relatives aux erreurs circulent dans un VI de la même façon que toutes les 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 **entrée d'erreur** et **sortie d'erreur** dans chaque VI que vous utilisez ou construisez pour transmettre les informations sur l'erreur par l'intermédiaire de votre 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 exécuter cette portion du code. Le nœud suivant fait de même, et ainsi de suite. LabVIEW signale l'erreur à la fin de l'exécution.

Clusters d'erreur

Les clusters **entrée d'erreur** et **sortie d'erreur** comprennent les éléments d'information suivants :

- **état** représente une valeur booléenne qui renvoie la valeur 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 **Erreur**. Reportez-vous à la section [Structures Condition](#) du chapitre 8, [Boucles et structures](#), pour obtenir de plus amples 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 des informations complémentaires 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 obtenir de plus amples 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 logiques de taille apte à faciliter la gestion. 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 obtenir de plus amples informations sur les sous-VIs.

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

Conception de projets avec plusieurs développeurs

Si plusieurs développeurs travaillent sur le même projet, définissez dès le départ des responsabilités pour la programmation, ainsi que des normes d'interfaces et de codage, pour assurer que le processus de développement et l'application fonctionnent ensemble de manière harmonieuse.

Reportez-vous au chapitre 6, *LabVIEW Style Guide*, du manuel *LabVIEW Development Guidelines*, pour obtenir de plus amples informations sur les normes de développement de code.

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 afficher les changements effectués entre différentes 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 obtenir de plus amples informations sur l'utilisation du contrôle du code source.

VIs modèles

Les modèles de VIs LabVIEW comprennent les sous-VIs, les fonctions, les structures et les objets de la face-avant dont vous avez besoin pour commencer à construire des applications de mesure courantes. Les modèles de VIs s'ouvrent comme VIs sans titre que vous devez enregistrer.

Sélectionnez **Fichier»Nouveau** pour afficher la boîte de dialogue **Nouveau**, qui comprend les modèles de VIs LabVIEW. Vous pouvez aussi afficher la boîte de dialogue **Nouveau** en sélectionnant **Nouveau** dans le menu déroulant **Nouveau** de la boîte de dialogue **LabVIEW**.

Si vous utilisez un modèle de VI comme sous-VI, LabVIEW vous demande de l'enregistrer comme VI avant de le fermer.

Création de VIs modèles

Vous pouvez créer des VIs modèles personnalisés pour éviter de mettre les mêmes composants sur la face-avant ou sur le diagramme chaque fois que vous voulez réaliser une opération similaire. Créez un VI modèle personnalisé en construisant un VI et en l'enregistrant comme modèle.

Autres types de document

Sélectionnez un élément sous l'en-tête **Autres types de document** dans la liste **Créer un nouvel objet** de la boîte de dialogue **Nouveau** pour commencer à créer des variables globales, des commandes personnalisées, des menus d'exécution, des VIs polymorphes ou des modèles de variables globales et de commandes.

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 obtenir de plus amples informations sur les sous-VIs.

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

LabVIEW inclut des centaines d'exemples de VIs que vous pouvez utiliser et incorporer dans les VIs que vous créez. Vous pouvez modifier un VI d'exemple pour qu'il s'adapte à votre application, ou vous pouvez copier et coller un ou plusieurs exemples dans le VI que vous créez.

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 pour commander 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 obtenir des informations complémentaires sur la construction des VIs de contrôle d'instruments.

Utilisez les VIs et fonctions d'acquisition de données pour acquérir des données à partir 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 obtenir de plus amples 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 *LabVIEW Analysis Concepts* pour obtenir de plus amples informations sur les concepts d'analyse et mathématiques de LabVIEW.

Construction de VIs qui accèdent à d'autres VIs

Utilisez les VIs et fonctions Contrôle d'applications pour contrôler la manière dont les VIs se comportent 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 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples 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 pour écrire des données dans d'autres applications ou pour lire des données provenant d'autres applications, comme 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 14, *E/S sur fichiers*, pour obtenir de plus amples informations sur la transmission de données en provenance ou à destination d'autres applications.

Utilisez les VIs et fonctions de communication pour transférer des données LabVIEW sur le Web à l'aide d'un protocole de communication tel que FTP et pour construire des applications client-serveur à l'aide de protocoles

de communication. Reportez-vous au chapitre 18, *Mise en réseau dans LabVIEW*, pour obtenir de plus amples informations sur la communication avec d'autres applications sur le réseau ou sur le Web.

(Windows) Utilisez les fonctions ActiveX pour ajouter des objets ActiveX aux VIs ou pour contrôler des applications ActiveX. Reportez-vous au chapitre 19, *Connectivité Windows*, pour obtenir de plus amples 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 échangent des données avec le diagramme du VI appelant. Cliquez sur l'icône **Sélectionner un VI** située sur la palette **Fonctions**, naviguez jusqu'à un VI et double-cliquez dessus ou placez-le sur le diagramme pour créer un appel de sous-VI à ce 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 obtenir de plus amples informations sur la configuration 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.

Lorsque LabVIEW appelle un sous-VI, celui-ci s'exécute en général sans afficher sa face-avant. Si vous voulez qu'une seule instance du sous-VI affiche sa face-avant lorsqu'il est appelé, effectuez un clic droit sur l'icône de ce sous-VI et sélectionnez **Configuration du nœud du sous-VI** dans le menu local. Si vous voulez que chaque instance du sous-VI affiche sa face-avant lorsqu'il est appelé, sélectionnez **Fichier»Propriétés du VI**, sélectionnez **Apparence de la fenêtre** dans le menu déroulant **Catégorie** et cliquez sur le bouton **Personnaliser**.

Recherche d'opérations courantes

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 représenté dans la figure 7-1 contient deux opérations identiques.

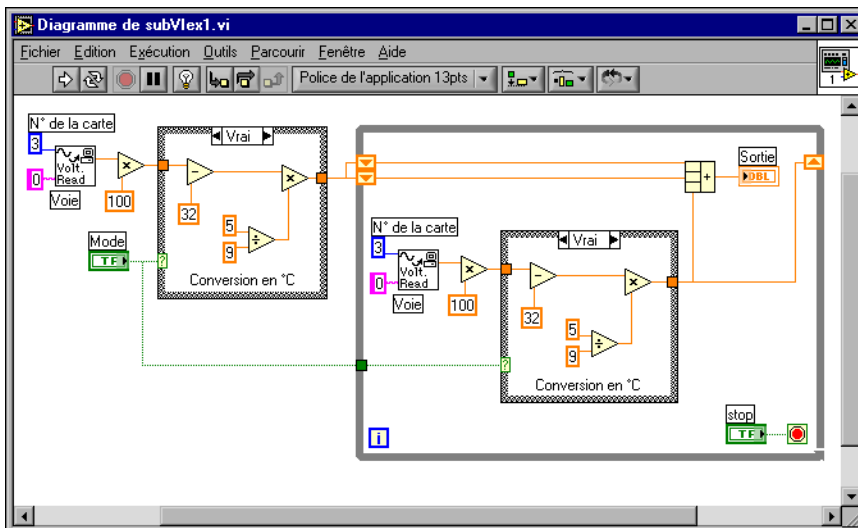


Figure 7-1. Diagramme avec deux opérations identiques

Vous pouvez créer un sous-VI qui effectue cette opération et appeler ce sous-VI deux fois, comme indiqué dans la figure 7-2.

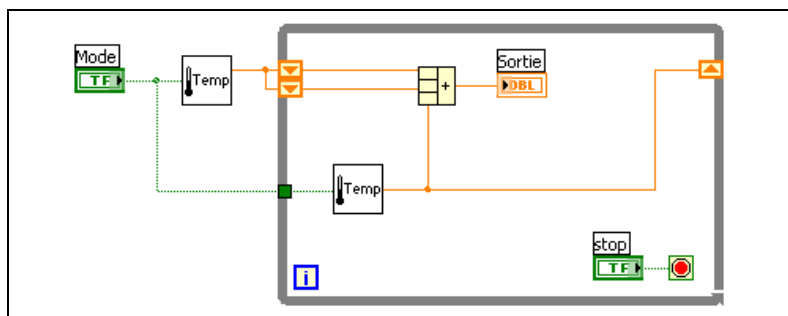


Figure 7-2. Appel d'un sous-VI à deux reprises

Vous pouvez réutiliser ce sous-VI dans d'autres VIs. Reportez-vous au chapitre 8, *Boucles et structures*, pour obtenir des informations complémentaires sur l'utilisation de boucles permettant de combiner des opérations courantes.

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 obtenir des informations complémentaires sur les connecteurs.

Définissez les connexions en affectant une commande ou un indicateur de face-avant à chaque terminal du connecteur. Pour définir un connecteur, cliquez avec le bouton droit sur l'icône dans le coin supérieur droit de la fenêtre de la face-avant et sélectionnez **Afficher le connecteur** dans le menu local pour afficher le connecteur. 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 par programmation, 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 10, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur le groupement des données au moyen de clusters.



Remarque L'affectation de plus de 16 terminaux à un VI risque de le rendre difficile à lire et à utiliser.

Sélectionnez un motif de terminal 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 7-3 présente des exemples de clusters d'erreur mal alignés et bien alignés.

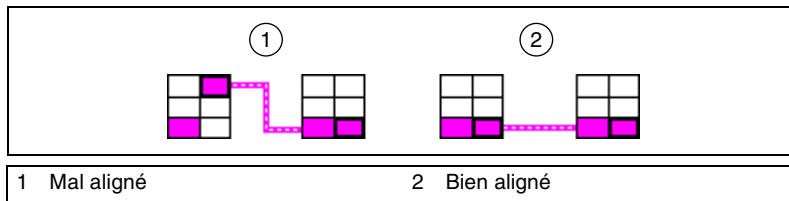


Figure 7-3. Clusters d'erreur mal aligné et bien aligné

Reportez-vous à la section *Connector Panes* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur le style à utiliser lorsque vous configurez un connecteur.

Définition des entrées et sorties nécessaires, recommandées et facultatives

Vous pouvez désigner quelles entrées et sorties sont nécessaires, recommandées et facultatives pour éviter que les utilisateurs n'oublient de câbler certains terminaux de sous-VI.

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

Pour les terminaux d'entrée, nécessaire signifie que le diagramme sur lequel vous placez le sous-VI sera brisé si vous ne câblez pas les entrées nécessaires. Nécessaire n'est pas une définition disponible pour les

terminaux de sortie. Pour les terminaux d'entrée et de sortie, recommandé ou facultatif signifie que le diagramme sur lequel vous placez le sous-VI peut s'exécuter même si vous ne câblez pas les terminaux recommandés ou facultatifs. Si vous ne câblez pas les terminaux, le VI ne génère pas de mise en garde.

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 étiquettes des terminaux nécessaires apparaissent en gras, alors que les étiquettes des terminaux recommandés apparaissent en texte normal et celles des terminaux facultatifs apparaissent grisées. Les étiquettes des terminaux facultatifs n'apparaissent pas si vous cliquez sur le bouton **Cacher les terminaux optionnels et le chemin complet** dans la fenêtre d'**Aide contextuelle**.

Création d'une icône



Chaque VI possède une icône, comme celle qui est représentée à gauche, qui s'affiche dans le coin supérieur droit de la fenêtre de la face-avant et du diagramme. Une icône est la représentation graphique d'un VI. Elle peut contenir du texte, des images ou les deux. Si vous utilisez un VI comme sous-VI, l'icône identifie le sous-VI sur le diagramme du VI de niveau principal.

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 32×32 pixels.

Selon le type d'écran 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.

Reportez-vous à la section *Icons* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la conception d'une icône.

Affichage de sous-VIs et de VIs Express sous forme d'icônes ou de nœuds extensibles

Vous pouvez afficher des VIs et des VIs Express comme icônes ou nœuds extensibles. Les nœuds extensibles apparaissent comme des icônes entourées par un champ coloré. Les sous-VIs apparaissent avec un champ jaune, et les VIs Express, avec un champ bleu. Utilisez les icônes si vous voulez économiser de l'espace sur le diagramme. Utilisez les nœuds extensibles pour rendre le câblage plus aisé et pour faciliter la documentation des diagrammes. Par défaut, les sous-VIs apparaissent sous forme d'icônes sur le diagramme et les VIs Express apparaissent sous forme de nœuds extensibles. Pour afficher un sous-VI ou un VI Express comme nœud extensible, cliquez avec le bouton droit sur le sous-VI ou le VI Express et sélectionnez **Afficher sous la forme d'une icône** dans le menu local pour désélectionner l'option.



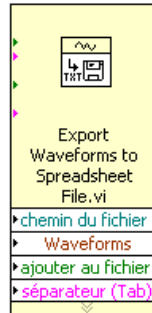
Remarque Si vous affichez un sous-VI ou un VI Express VI sous forme de nœud extensible, vous ne pouvez pas afficher les terminaux de ce nœud, ni autoriser l'accès à la base de données pour ce nœud.

Quand vous redimensionnez un sous-VI ou un VI Express extensible, les terminaux d'entrée et de sortie du sous-VI ou du VI Express apparaissent au-dessous de l'icône. Les terminaux facultatifs apparaissent sur fond gris. Les terminaux d'entrée ou de sortie recommandés ou nécessaires que vous n'affichez pas apparaissent sous forme de flèches d'entrée ou de sortie dans le champ coloré qui entoure l'icône du sous-VI ou du VI Express. Si vous câblez un fil de liaison à un terminal facultatif lorsque le sous-VI ou le VI Express est développé, redimensionnez le sous-VI ou le VI Express de sorte que le terminal facultatif ne soit plus affiché dans le champ développé et que le terminal facultatif apparaisse sous forme de flèche d'entrée ou de sortie dans le champ coloré. Cependant, si vous déconnectez le terminal facultatif, la flèche d'entrée ou de sortie n'apparaît plus.

Par défaut, les entrées apparaissent au-dessus des sorties lorsque vous développez le sous-VI ou le VI Express. Effectuez un clic droit sur un terminal dans le champ redimensionnable et sélectionnez **Sélectionner une entrée/sortie** dans le menu local afin de sélectionner l'entrée ou la sortie à afficher. Un trait sépare les entrées et les sorties dans le menu local. Les étiquettes des sous-VIs et des VIs Express extensibles apparaissent dans le champ coloré qui entoure l'icône. Pour redimensionner le nœud extensible

de sorte qu'il puisse contenir le nom de chaque terminal sur une seule ligne dans le champ redimensionnable, cliquez avec le bouton droit sur le sous-VI ou le VI Express et sélectionnez **Ajuster au texte** dans le menu local.

Le sous-VI extensible suivant affiche entre quatre et 10 terminaux d'entrée et de sortie.



Reportez-vous au manuel *Initiation à LabVIEW* pour obtenir des informations complémentaires sur les VIs Express.

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

Convertissez une portion de 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 commode de créer un sous-VI à partir d'une sélection, mais une planification est nécessaire pour créer une hiérarchie de VIs logique. Envisagez quels objets devront être inclus dans la sélection et évitez de changer la fonctionnalité du VI résultant.

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 l'afficher 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 obtenir de plus amples informations sur la configuration 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 **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»Afficher la hiérarchie du VI** pour afficher la fenêtre **Hiérarchie**. Utilisez cette fenêtre pour afficher les sous-VIs et les autres nœuds constituant le VI actuel.

Lorsque vous déplacez l'outil Doigt sur les objets dans la fenêtre **Hiérarchie**, LabVIEW affiche le nom de chaque VI. Vous pouvez utiliser l'outil Flèche pour déplacer un VI de la fenêtre **Hiérarchie** vers le diagramme afin d'utiliser le VI comme 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 **Hiérarchie** pour afficher la face-avant de ce VI.

Un VI qui contient des sous-VIs possède un bouton flèche sur son cadre inférieur. 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 `.lib`. National Instruments vous recommande d'enregistrer les VIs dans des fichiers individuels organisés en répertoires, particulièrement si plusieurs développeurs travaillent sur le même projet. Reportez-vous à la section *Organizing VIs in Directories* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur l'organisation des VIs en répertoires.

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 dans 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.
(Mac OS) Mac OS 9.x ou version antérieure 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.
- Il est plus facile de transférer une bibliothèque de VIs à d'autres plates-formes que de transférer plusieurs VIs individuellement. Vous pouvez aussi vous assurer que l'utilisateur reçoit tous les fichiers nécessaires.

- 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.

(Windows 2000/XP/Me/98) Vous pouvez aussi double-cliquer sur un fichier `.llb` dans l'explorateur Windows pour afficher le contenu d'une bibliothèque afin de pouvoir ouvrir, déplacer, copier, renommer ou supprimer des fichiers de la bibliothèque.

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 `#`.

Utilisez des noms de VIs de moins de 31 caractères si les utilisateurs sont susceptibles d'exécuter les VIs sur Mac OS 9.x ou version antérieure.

Enregistrement au format antérieur

Vous pouvez enregistrer des VIs dans un format de version antérieure pour faciliter le passage d'une version de LabVIEW à une autre et maintenir les VIs dans deux versions de LabVIEW lorsque cela est 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.

Si vous souhaitez distribuer à d'autres utilisateurs LabVIEW des VIs qui comprennent le code source du diagramme, vous pouvez attribuer une protection par mot de passe aux diagrammes. Le diagramme est toujours disponible, mais les utilisateurs doivent entrer un mot de passe pour l'afficher ou l'éditer.

Si vous voulez distribuer des VIs à des développeurs d'autres langages de programmation, vous pouvez construire une application autonome 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.



Remarque Vous pouvez créer des applications autonomes ou des bibliothèques partagées uniquement dans les systèmes de développement professionnel LabVIEW ou en utilisant l'Application Builder.

Une autre option de distribution des VIs consiste à supprimer le code source du diagramme de façon à ce que d'autres utilisateurs puissent éditer le VI. 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.

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 bibliothèque partagée (DLL)** pour utiliser l'Application Builder afin de créer des applications autonomes et des installeurs ou des bibliothèques partagées (DLLs) pour des VIs. Utilisez les bibliothèques partagées si vous voulez appeler les VIs dans la bibliothèque partagée à l'aide de langages de programmation textuels, comme LabWindows™/CVI™, Microsoft Visual C++ et Microsoft Visual Basic.

Les bibliothèques partagées sont utiles pour partager les fonctionnalités de vos VIs avec d'autres développeurs. L'utilisation de bibliothèques partagées permet aux langages de programmation autres que LabVIEW d'accéder au code développé avec LabVIEW.

D'autres développeurs peuvent exécuter l'application autonome ou utiliser les bibliothèques partagées, mais ils ne peuvent ni afficher ni modifier les diagrammes.



Remarque La version Système de développement professionnel LabVIEW inclut l'Application Builder. Si vous utilisez la version de base LabVIEW ou le Système de développement complet LabVIEW, vous pouvez acheter l'Application Builder séparément en visitant le site Web de National Instruments à ni.com/info et en entrant le code d'information rd1v21. Utilisez les onglets de la boîte de dialogue **Construire une application ou une bibliothèque partagée (DLL)** pour configurer divers paramètres pour l'application ou la bibliothèque partagée que vous voulez construire. Après avoir défini ces paramètres, enregistrez-les dans un script pour pouvoir reconstruire l'application rapidement, si nécessaire.

Reportez-vous au manuel *Guide de l'utilisateur de l'Application Builder de LabVIEW* pour obtenir de plus amples 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 et structures*, décrit comment utiliser les boucles et 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, *Programmation événementielle*, décrit comment enregistrer des événements de manière dynamique et de manière statique, et comment créer et nommer vos propres événements.
- Le chapitre 10, *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 11, *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 12, *Graphes et graphes déroulants*, explique comment utiliser les graphes et les graphes déroulants pour afficher des courbes de données sous forme graphique.
- Le chapitre 13, *VIs Graphisme et son*, explique comment afficher ou modifier des graphiques ou des sons dans des VIs.

- Le chapitre 14, *E/S sur fichiers*, explique comment effectuer des opérations d'E/S sur fichiers.
- Le chapitre 15, *Documentation des VIs*, explique comment documenter et imprimer des VIs.
- Le chapitre 16, *Personnalisation des VIs*, explique comment configurer des VIs et des sous-VIs pour les adapter aux besoins de vos applications.
- Le chapitre 17, *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 18, *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 19, *Connectivité Windows*, explique comment fournir un ensemble public d'objets, de commandes et de fonctions auxquels d'autres applications de Windows peuvent avoir accès.
- Le chapitre 20, *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 DLLs.
- Le chapitre 21, *Formules et équations*, explique comment utiliser des équations dans les VIs.

Boucles et structures

Les structures sont des représentations graphiques de boucles et de conditions dans les langages de programmation textuels. Utilisez des structures dans votre diagramme pour répéter des blocs de code et pour exécuter le code de manière conditionnelle ou dans un ordre spécifique.

Tout comme les autres nœuds, les structures possèdent des terminaux qui les connectent à d'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 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 obtenir des informations complémentaires sur l'utilisation des structures.

Utilisez les structures suivantes de la palette **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 21, *Formules et équations*, pour obtenir de plus amples 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 boucles For et boucles While

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

Boucles For



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

La valeur du terminal de décompte (terminal d'entrée), représenté à 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 obtenir de plus amples informations sur le paramétrage implicite du décompte.



Le terminal d'itération (terminal de sortie), représenté à gauche, contient le nombre d'itérations achevées. Le décompte des itérations démarre toujours à zéro. Pendant la première itération, le terminal d'itération renvoie 0.

Les terminaux de décompte et d'itération sont tous les deux des entiers signés 32 bits. 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 ou un nombre négatif au terminal de décompte, la boucle ne s'exécute pas et les sorties contiennent les données par défaut pour ce type de données.

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 et Nœud de rétroaction dans les boucles* de ce chapitre pour obtenir de plus amples informations sur l'ajout de registres à décalage à une boucle.

Boucles While



Tout comme la boucle “do” ou la boucle “repeat-until” (répéter jusqu’à...) dans les langages de programmation textuels, la boucle While, représentée à 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 **Arrêter sur condition Vraie**, représenté à gauche. Lorsqu’un terminal de condition d’arrêt est défini sur **Arrêter sur condition Vraie**, la boucle While exécute son sous-diagramme jusqu’à ce que le terminal condition d’arrêt reçoive une valeur VRAI. 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 **Continuer sur condition Vraie**, représenté à gauche, à partir du menu local. Lorsqu’un terminal conditionnel est défini 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 aussi utiliser l’outil Doigt pour cliquer sur le terminal conditionnel afin de changer la condition.



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. En outre, 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 [Détection et gestion d’erreur](#) du chapitre 6, [Exécution et mise au point des VIs](#), pour obtenir de plus amples informations sur la gestion et les clusters d’erreur.



Le terminal d’itération (terminal de sortie), représenté à gauche, contient le nombre d’itérations achevées. Le décompte des itérations démarre toujours à zéro. Pendant la première itération, le terminal d’itération renvoie 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 et Nœud de rétroaction dans les boucles](#) de ce chapitre pour obtenir de plus amples 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, et que la commande est définie à la valeur FAUX si 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 hors de la boucle est égale à VRAI et que le terminal conditionnel est défini à **Continuer sur condition Vraie**.

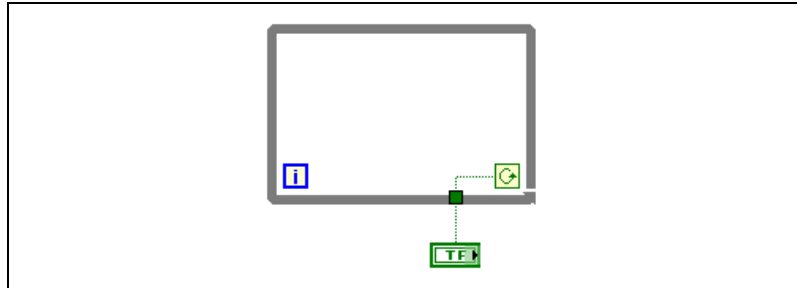


Figure 8-1. Boucle While infinie

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 abandonner le VI en cliquant sur le bouton **Abandonner** 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 10, [Groupement des données au moyen de chaînes, de tableaux et de clusters](#), pour obtenir de plus amples 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 à raison 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 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.

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 transmise à la boucle. Vous pouvez empêcher que la valeur par défaut ne soit transmise à la boucle While en utilisant la fonction Taille d'un tableau. La fonction Taille d'un tableau indique combien d'éléments figurent dans le tableau. Définissez 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.

Utilisation de boucles pour construire des tableaux

Vous pouvez non seulement utiliser les boucles pour lire et traiter les éléments d'un tableau, mais aussi utiliser les boucles For et While pour construire ces tableaux. Câblez la sortie d'un VI ou d'une fonction dans la boucle au bord de la boucle. Si vous utilisez une boucle While, effectuez un clic droit sur le tunnel qui résulte et sélectionnez **Activer l'indexation** dans le menu local. Sur la boucle For, l'indexation est activée par défaut. La sortie du tunnel est un tableau contenant chaque valeur que le VI ou la fonction renvoie après chaque itération de la boucle.

Reportez-vous à `exemples\general\arrays.llb` pour obtenir des exemples de construction de tableaux.

Registres à décalage et Nœud de rétroaction dans les boucles

Utilisez les registres à décalage ou le nœud de rétroaction sur les boucles For ou While pour transférer des valeurs d'une itération de la boucle à la suivante.

Registres à décalage



Utilisez les registres à décalage lorsque vous voulez transmettre des valeurs d'itérations précédentes par l'intermédiaire de la boucle. Un registre à décalage apparaît sous la forme d'une paire de terminaux, représentés à 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.

Initialisez un registre à décalage en câblant une commande ou une constante au terminal du registre à décalage sur le côté gauche de la boucle. L'initialisation d'un registre à décalage réinitialise la valeur de départ que le registre à décalage transmet lors du premier tour de boucle lorsque le VI s'exécute. Si vous n'initialisez pas le registre, la boucle utilise la valeur écrite sur le registre lors de la dernière exécution de la boucle ou la valeur par défaut du type de données si la boucle ne s'est pas encore 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. Après l'exécution de la boucle, la dernière valeur stockée dans le registre à décalage est renvoyée en sortie par le terminal de droite.

Vous pouvez ajouter plus d'un registre à décalage à une boucle. Si votre boucle contient plusieurs opérations qui utilisent des valeurs d'itérations précédentes, utilisez plusieurs registres à décalage pour stocker les valeurs de données résultant des différents traitements qui ont lieu dans la structure.

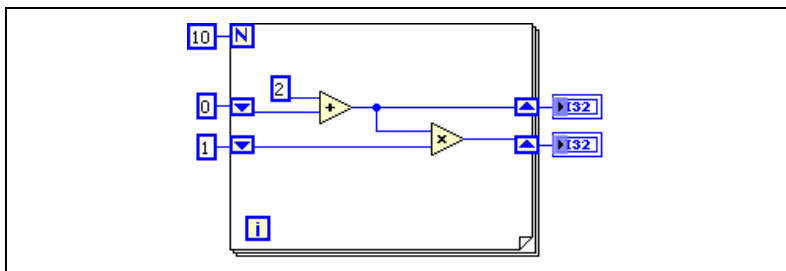


Figure 8-2. Registres à décalage multiples

Registres à décalage empilés

Les registres à décalage vous permettent d'accéder aux données provenant d'itérations précédentes de la boucle. Pour créer un registre à décalage empilé, cliquez avec le bouton droit sur le terminal gauche et sélectionnez **Ajouter un élément** dans le menu local. Les registres à décalage empilés conservent en mémoire les valeurs de multiples itérations précédentes et transmettent ces valeurs aux itérations suivantes.

Les registres à décalage empilés ne peuvent avoir lieu que sur le côté gauche de la boucle car le terminal de droite transfère uniquement les données générées de l'itération en cours à l'itération suivante.

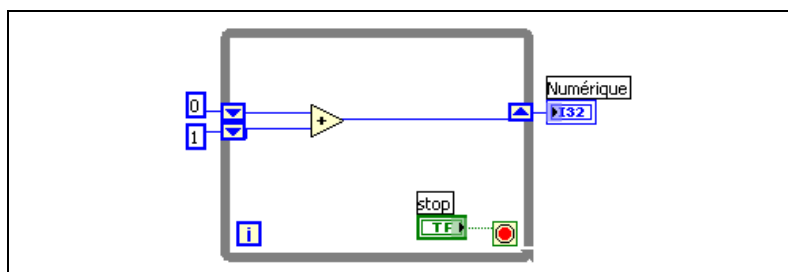


Figure 8-3. Registres à décalage empilés

Si vous ajoutez un autre élément au terminal de gauche, les valeurs des deux dernières itérations sont transmises à l'itération suivante, la valeur d'itération la plus récente étant stockée dans le registre à décalage supérieur. Le terminal du bas stocke les données qui lui sont transmises de l'itération précédente.

Remplacement des registres à décalage par des tunnels

Pour remplacer des registres à décalage par des tunnels, cliquez avec le bouton droit sur le registre à décalage et sélectionnez **Remplacer par des tunnels** dans le menu local lorsque vous ne voulez plus transférer de valeurs d'une itération de boucle à la suivante.

Si vous remplacez un registre à décalage de sortie par un tunnel sur une boucle For, le fil de liaison câblé à des nœuds hors de la boucle se brise car la boucle For active l'indexation par défaut. Effectuez un clic droit sur le tunnel et sélectionnez **Désactiver l'indexation à la source** dans le menu local afin de désactiver l'indexation et de corriger automatiquement le fil de liaison brisé. Si vous voulez activer l'indexation, supprimez le fil de liaison brisé et le terminal d'indicateur, effectuez un clic droit sur le tunnel et sélectionnez **Créer un indicateur**.

Reportez-vous à la section [Auto-indexation des boucles](#) de ce chapitre pour obtenir des informations complémentaires sur l'indexation dans les boucles.

Remplacement des tunnels par des registres à décalage

Pour remplacer des tunnels par des registres à décalage, cliquez avec le bouton droit sur le tunnel et sélectionnez **Remplacer par un registre à décalage** dans le menu local lorsque vous voulez transférer des valeurs d'une itération de boucle à la suivante. S'il n'y a aucun tunnel sur le cadre de la boucle à l'opposé du tunnel sur lequel vous avez effectué un clic droit, LabVIEW crée automatiquement une paire de terminaux de registres à décalage. S'il existe un tunnel sur le cadre de la boucle à l'opposé du tunnel sur lequel vous avez effectué un clic droit, LabVIEW remplace ce dernier par un terminal de registre à décalage et le curseur se transforme en icône de registre à décalage. Cliquez sur un tunnel sur le côté opposé de la boucle pour remplacer ce tunnel par un registre à décalage ou cliquez sur le diagramme pour placer le registre à décalage sur le cadre de la boucle directement à l'opposé de l'autre terminal de registre à décalage. Si le registre à décalage apparaît derrière un tunnel, c'est qu'il n'est pas câblé.

Si vous convertissez un tunnel dont l'indexation est activée en un registre à décalage sur une boucle While, le fil de liaison connecté à un nœud hors de la boucle se brise car les registres à décalage ne permettent pas l'indexation. Supprimez le fil de liaison brisé, connectez le fil de liaison de sortie situé

du côté droit du registre à décalage à un autre tunnel, cliquez avec le bouton droit sur le tunnel et sélectionnez **Activer l'indexation** dans le menu local, puis câblez le tunnel au nœud.

Reportez-vous à la section *Auto-indexation des boucles* de ce chapitre pour obtenir des informations complémentaires sur l'indexation dans les boucles.

Nœud de rétroaction



Le nœud de rétroaction, représenté à gauche, apparaît automatiquement uniquement dans une boucle For ou While chaque fois que vous câblez la sortie d'un sous-VI, d'une fonction ou d'un groupe de sous-VIs ou de fonctions à l'entrée de ce même VI, fonction ou groupe. Tout comme le registre à décalage, le nœud de rétroaction stocke les données dès que la boucle termine une itération, envoie cette valeur à l'itération suivante de la boucle et transfère le type de données. Utilisez le nœud de rétroaction pour éviter des fils de liaison d'une longueur superflue dans les boucles. La flèche du nœud de rétroaction indique le sens dans lequel les données circulent le long du fil de liaison.

Vous pouvez aussi sélectionner le nœud de rétroaction et le placer à l'intérieur d'une boucle For ou While uniquement. Si vous placez le nœud de rétroaction sur le fil de liaison avant d'avoir créé une branche pour le fil de liaison qui connecte les données au tunnel, le nœud de rétroaction transmet chaque valeur au tunnel. Si vous placez le nœud de rétroaction sur le fil de liaison après avoir créé une branche pour le fil de liaison qui connecte les données au tunnel, le nœud de rétroaction retransmet chaque valeur à l'entrée du VI ou de la fonction et transmet ensuite la dernière valeur au tunnel.

Par exemple, la boucle For de la figure 8-4 réalise 10 itérations. Le nœud de rétroaction transmet la valeur de l'itération précédente de la boucle au tunnel avant de transmettre cette valeur à l'entrée de la fonction Additionner. La valeur dans le tunnel est toujours celle qui provient de l'itération précédente. Sur la dernière itération de la boucle, le nœud de rétroaction garde la dernière valeur, soit 45 dans ce cas, mais ne la transmet pas au tunnel ni à l'indicateur numérique. Quand le VI termine son exécution, la valeur dans l'indicateur numérique est 36, c'est-à-dire la valeur de l'itération précédente et non la dernière itération de la boucle.

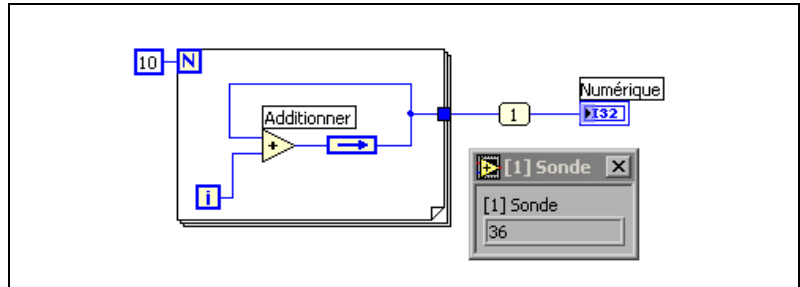


Figure 8-4. Transmission de l'avant-dernière valeur en sortie de la boucle For

La boucle For de la figure 8-5 accomplit aussi 10 itérations. Toutefois, le nœud de rétroaction transmet la valeur de l'itération précédente uniquement à l'entrée de la fonction Additionner à chaque itération de la boucle. Sur la dernière itération de la boucle, le nœud de rétroaction transmet la valeur de l'itération précédente (36) à l'entrée de la fonction Additionner. La fonction Additionner ajoute la valeur générée par le terminal d'itération (9) à la valeur transmise par le nœud de rétroaction (36) et envoie ensuite le résultat au tunnel. Lorsque le VI termine son exécution, la valeur affichée par l'indicateur numérique est **45**.

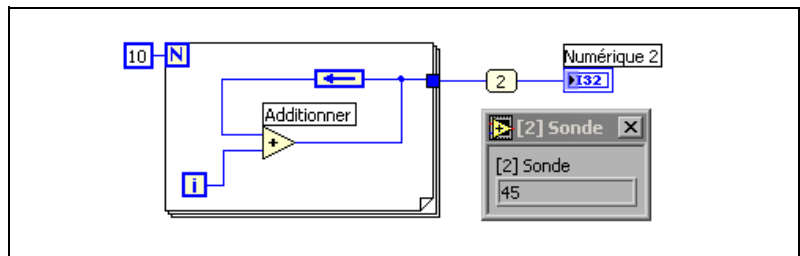


Figure 8-5. Transmission de la dernière valeur en sortie de la boucle For

Initialisation des nœuds de rétroaction

Cliquez avec le bouton droit sur le nœud de rétroaction et sélectionnez **Terminal d'initialisation** dans le menu local pour ajouter le terminal d'initialisation au cadre de la boucle afin d'initialiser celle-ci. Si vous sélectionnez le nœud de rétroaction ou si vous convertissez un registre à décalage initialisé en nœud de rétroaction, la boucle s'affiche automatiquement avec un terminal d'initialisation. L'initialisation d'un nœud de rétroaction réinitialise la valeur de départ que le nœud de rétroaction transmet lors du premier tour de boucle lorsque le VI s'exécute. Si vous n'initialisez pas le nœud de rétroaction, il transmet la dernière valeur écrite au nœud ou la valeur par défaut du type de données si la boucle

n'a jamais été exécutée. Si vous laissez l'entrée du terminal d'initialisation non câblé, chaque fois que vous exécutez le VI, l'entrée initiale du nœud est la valeur provenant de l'exécution précédente.

Remplacement de registres à décalage par un nœud de rétroaction

Pour remplacer un registre à décalage par un nœud de rétroaction, cliquez avec le bouton droit sur le registre à décalage et sélectionnez **Remplacer par un nœud de rétroaction** dans le menu local. Pour remplacer un nœud de rétroaction par des registres à décalage, cliquez avec le bouton droit sur le nœud de rétroaction et sélectionnez **Remplacer par un registre à décalage** dans le menu local.

Contrôle de la vitesse

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 patienter pendant une période de temps exprimée en millisecondes avant que la boucle ne s'exécute à nouveau.

Reportez-vous à la section *Memory and Speed Optimization* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur l'utilisation d'une fonction d'attente dans une boucle afin d'optimiser l'utilisation de la mémoire.

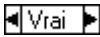
Structures Condition et Séquence

Les structures Condition, Séquence empilée, Séquence déroulée et les structures Événement contiennent de multiples sous-diagrammes. Une structure Condition exécute un sous-diagramme selon la valeur d'entrée transmise à la structure. Une structure Séquence empilée et une structure Séquence déroulée exécutent tous leurs sous-diagrammes dans l'ordre séquentiel. Une structure Événement exécute ses sous-diagrammes selon les interactions de l'utilisateur avec le VI.

Structures Condition



Une structure Condition, représentée à 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 figurant en haut de la structure Condition, représenté à gauche, contient le nom de la valeur du sélecteur qui correspond à 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 à gauche, pour déterminer la condition qui s'exécute. Vous devez câbler un entier, un booléen, une chaîne ou un type énuméré au terminal de sélection. Vous pouvez positionner le terminal de sélection n'importe où sur le bord gauche de la structure Condition. Si le type de donnée du terminal de sélection est booléen, la structure a un cas `Vrai` et un cas `Faux`. Si le terminal de sélection est un entier, une chaîne de caractères ou un type énuméré, la structure peut avoir n'importe quel nombre de cas.

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 de type entier et que vous spécifiez des conditions pour les valeurs 1, 2 et 3, vous devez spécifier une condition par défaut qui s'exécutera si la valeur d'entrée est 4 ou n'importe quelle autre valeur entière valide.

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. Dans les listes, séparez les valeurs par des virgules. Dans les gammes numériques, spécifiez la gamme sous la forme 10..20, ce qui signifie tous les nombres entre 10 et 20. Vous pouvez également utiliser des gammes ouvertes. Par exemple, ..100 représente tous les nombres inférieurs ou égaux à 100, et 100.. représente tous les nombres supérieurs ou égaux à 100. Vous pouvez aussi utiliser une combinaison de listes et de gammes, comme par exemple ..5, 6,

7. .10, 12, 13, 14. Si vous entrez des valeurs contenant 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 s'affiche maintenant sous la forme **. .10, 12. .14**. Dans les gammes de chaînes, la gamme a. . c inclut la totalité de a et de b, mais pas c. Une gamme de la forme a. . c, c inclut la valeur finale de 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 obtenir la liste de ces codes à 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 aussi cliquer avec le bouton droit sur le tunnel de sortie et sélectionner **Utiliser la valeur par défaut si non câblé** dans le menu local pour utiliser la valeur par défaut du type de données du tunnel pour tous les tunnels non câblés.

Utilisation des structures Condition pour la gestion d'erreur

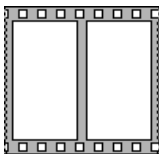
Lorsque vous câblez un cluster d'erreur au terminal de sélection d'une structure Condition, le sélecteur de condition affiche deux conditions, Erreur et Pas d'erreur, et le cadre de la structure Condition prend la couleur rouge pour Erreur et verte pour Pas d'erreur. La structure Condition exécute le 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 obtenir de plus amples informations sur la gestion des erreurs.

Structures Séquence

La structure Séquence, représentée à gauche, contient un ou plusieurs sous-diagrammes, ou étapes, qui s'exécutent dans l'ordre séquentiel. Les structures Séquence ne sont pas couramment utilisées dans LabVIEW. Reportez-vous aux sections *Utilisation des structures Séquence* et *Éviter un usage excessif des structures Séquence* de ce chapitre pour obtenir de plus amples informations sur les situations dans lesquelles il convient d'utiliser des structures séquence.

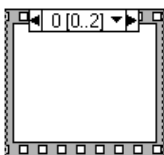
Il existe deux types de structures Séquence : la structure Séquence déroulée et la structure Séquence empilée.

Structure Séquence déroulée

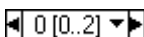


La structure Séquence déroulée, représentée à gauche, affiche toutes les étapes à la fois et exécute les étapes depuis la gauche vers la droite jusqu'à l'exécution de la dernière étape. Utilisez la structure Séquence déroulée pour éviter d'avoir à utiliser des variables locales de séquence et pour mieux documenter votre diagramme. Lorsque vous ajoutez ou supprimez des étapes dans une structure Séquence déroulée, la structure se redimensionne automatiquement. Pour réorganiser les étapes d'une structure Séquence déroulée, effectuez des couper-coller d'une étape à une autre.

Structure Séquence empilée



La structure Séquence empilée, représentée à gauche, empile chaque étape pour que vous puissiez voir une seule étape à la fois et exécute l'étape 0, puis l'étape 1, et ainsi de suite jusqu'à l'exécution de la dernière étape. La structure Séquence empilée renvoie des données uniquement après que la dernière étape s'est exécutée. Utilisez la structure Séquence empilée si vous voulez économiser de l'espace sur votre diagramme.



L'identificateur de sélecteur de séquence, représenté à gauche, en haut de la structure Séquence empilée, contient le numéro de l'étape en cours et la gamme des étapes. Utilisez le sélecteur de séquence pour naviguer parmi les cadres disponibles et les réorganiser. L'étiquette de l'étape située au sommet de la structure Séquence empilée est semblable 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'incrémentement 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. Cliquez avec le bouton droit sur le bord d'un cadre, sélectionnez **Changer cette étape en**, puis sélectionnez un numéro d'étape dans le menu local pour réorganiser l'ordre de la structure Séquence empilée.

Contrairement au sélecteur de condition, vous ne pouvez pas entrer de valeurs dans l'étiquette du cadre. Quand vous ajoutez, supprimez ou réarrangez des étapes dans une structure Séquence empilée, LabVIEW ajuste automatiquement les numéros qui figurent dans les étiquettes des étapes.

Utilisation des structures Séquence

Utilisez les structures 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 étape 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 artificielle des données* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur la dépendance des données.

Les tunnels des structures Séquence empilée 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 empilée uniquement lorsque toutes les étapes ont terminé leur exécution et non quand chaque étape a terminé son exécution. Comme pour les structures Condition, les données câblées aux tunnels d'entrée sont disponibles pour tous les cadres.



Pour transmettre des données d'une étape à l'étape suivante dans une structure Séquence empilée, utilisez un terminal de variable locale de séquence, représenté à gauche. Une flèche pointant vers l'extérieur apparaît dans le terminal de variable locale de séquence sur l'étape qui contient la source de données. Le terminal dans les étapes ultérieures 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 étapes se trouvant avant le premier cadre dans lequel vous avez câblé la variable locale de séquence.

Reportez-vous à `exemples\general\structs.llb` pour obtenir 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 les 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 obtenir de plus amples informations sur les E/S d'erreurs.

Par ailleurs, n'utilisez pas de structures Séquence si vous voulez mettre un indicateur à jour à partir de différentes étapes 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'une étape différente, vous ne pouvez pas mettre l'indicateur à jour à partir de chaque étape, comme indiqué par le fil de liaison brisé dans la structure Séquence empilée de la figure 8-6.

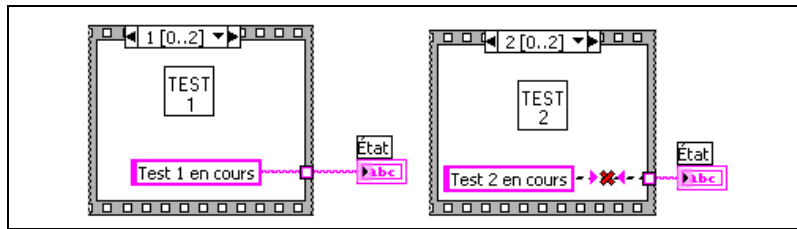


Figure 8-6. Mise à jour d'un indicateur à partir de différentes étapes dans une structure Séquence empilée

Étant donné que toutes les étapes de la structure Séquence empilée s'exécutent avant toute transmission de données, une seule étape peut attribuer une valeur à l'indicateur **état**.

Utilisez à la place une structure Condition et une boucle While, comme l'indique la figure 8-7.

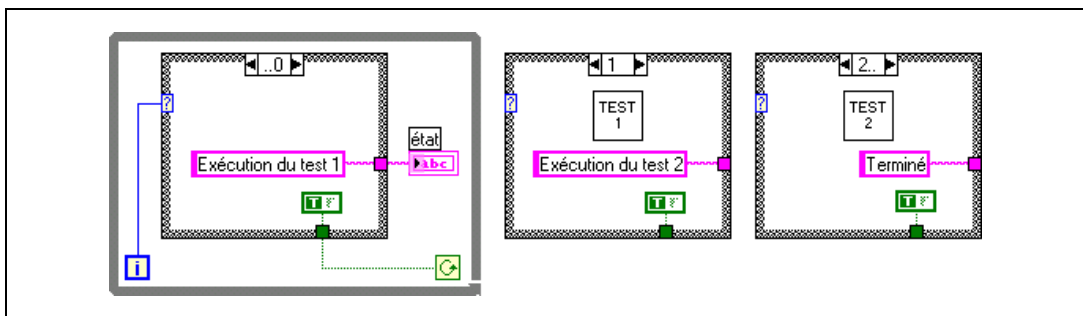


Figure 8-7. Mise à jour d'un indicateur à partir de différentes conditions dans une structure Condition

Chaque condition de la structure Condition est équivalente à une étape 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 toutes ses étapes, même si une erreur se produit.

Remplacement de structures Séquence

Cliquez avec le bouton droit sur une structure Séquence déroulée et sélectionnez **Remplacer par une Séquence empilée** dans le menu local pour convertir une structure Séquence déroulée en structure Séquence empilée. Cliquez avec le bouton droit sur une structure Séquence empilée et sélectionnez **Remplacer»Remplacer par une Séquence déroulée** dans le menu local pour convertir une structure Séquence empilée en structure Séquence déroulée.

Programmation événementielle

LabVIEW est un environnement de programmation dans lequel le flux de données détermine l'ordre d'exécution des éléments du diagramme. Ses fonctionnalités de programmation événementielle étendent l'environnement par flux de données de LabVIEW pour permettre une interaction directe de l'utilisateur avec la face-avant et d'autres activités asynchrones pour mieux contrôler l'exécution du diagramme.



Remarque Les fonctionnalités de la programmation événementielle sont uniquement disponibles dans les versions Systèmes de développement complet et professionnel de LabVIEW. Vous pouvez exécuter un VI construit avec ces fonctionnalités avec la version de base de LabVIEW, mais vous ne pouvez pas reconfigurer les composants de la gestion d'événement.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations complémentaires sur l'utilisation des événements dans votre application.

Que sont les événements ?

Un événement est une notification asynchrone vous informant que quelque chose a eu lieu. Les événements peuvent être générés dans l'interface utilisateur, par des E/S externes ou par d'autres parties du programme. Les événements d'interface utilisateur comprennent les clics de souris, la saisie par les touches du clavier et ainsi de suite. Les événements d'E/S externes comprennent les compteurs ou les déclenchements matériels qui avertissent lorsque l'acquisition de données est achevée ou qu'une erreur s'est produite. D'autres types d'événement peuvent être générés par programmation et utilisés pour communiquer avec différentes parties du programme. LabVIEW prend en charge les événements de l'interface utilisateur et ceux qui sont générés par programmation, mais pas les événements des E/S externes.

Dans une programmation événementielle, les événements qui ont lieu dans le système exercent une influence directe sur le déroulement de l'exécution. Par contraste, une programmation basée sur des procédures s'exécute dans un ordre séquentiel prédéterminé. Les programmes événementiels comprennent en général une boucle qui attend qu'un événement se produise, exécute le code en réponse à cet événement et effectue une nouvelle itération en attendant l'événement suivant. La manière dont le programme répond à chaque événement dépend du code qui a été rédigé pour cet événement particulier. L'ordre dans lequel le programme commandé par événements s'exécute dépend des événements qui ont lieu et de l'ordre dans lequel ils se produisent. Il est possible que certaines sections du programme s'exécutent fréquemment du fait que les événements qu'elles attendent se produisent fréquemment, alors que d'autres sections du programme peuvent ne pas s'exécuter du tout du fait que les événements attendus ne se produisent pas.

Pour quelle raison utiliser les événements ?

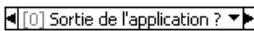
Utilisez des événements d'interface utilisateur dans LabVIEW pour synchroniser les actions utilisateur sur la face-avant avec l'exécution du diagramme. Les événements vous permettent d'exécuter une condition de gestion d'événement spécifique chaque fois qu'un utilisateur effectue une action particulière. Sans les événements, le diagramme est obligé d'interroger l'état des objets de la face-avant dans une boucle, pour vérifier si un changement a eu lieu. L'interrogation de la face-avant consomme une quantité importante de temps processeur et peut manquer de détecter des changements s'ils se produisent trop rapidement. L'utilisation d'événements pour répondre à des actions utilisateur particulières élimine la nécessité d'interroger la face-avant pour savoir quelles actions l'utilisateur a réalisées. Au lieu de cela, LabVIEW avertit directement le diagramme chaque fois que l'interaction que vous avez spécifiée se produit. L'utilisation d'événements réduit les exigences en temps processeur du programme, simplifie le code du diagramme et garantit que le diagramme est en mesure de répondre à toutes les interactions provenant de l'utilisateur.

Utilisez des événements générés par programmation pour communiquer avec différentes parties du programme qui ne dépendent pas du déroulement de l'exécution. Les événements générés par programmation offrent pratiquement les mêmes avantages que les événements d'interface utilisateur, avec lesquels ils peuvent partager le même code de gestion d'événements, ce qui simplifie la tâche d'implémentation d'architectures avancées, comme les machines d'état placées en files d'attente sur la base d'événements.

Composantes de la structure Événement



Utilisez la structure Événement, représentée à gauche, pour gérer des événements dans un VI. La structure Événement fonctionne de la même manière qu'une structure Condition avec une fonction intégrée Attendre une notification. La structure Événement peut avoir plusieurs conditions, correspondant chacune à un sous-programme distinct de gestion d'événement. Vous pouvez configurer chaque condition pour gérer un ou plusieurs événements, mais un seul de ces événements peut avoir lieu à la fois. Lorsque la structure Événement s'exécute, elle attend que l'un des événements configurés se produise, puis elle exécute la condition correspondant à cet événement. La structure Événement termine son exécution après avoir géré un seul événement. Elle n'effectue pas d'itération de boucle implicitement pour gérer plusieurs événements. Tout comme la fonction Attendre une notification, la structure Événement peut être mise hors délai pendant qu'elle attend une notification d'événement. Si cela se produit, une condition de type Timeout spécifique s'exécute.



L'étiquette du sélecteur d'événement en haut de la structure Événement, représentée à gauche, indique quels événements entraînent l'exécution de la condition qui est actuellement affichée. Pour afficher d'autres conditions d'événement, cliquez sur la flèche vers le bas contre le nom de la condition et sélectionnez une autre condition dans le menu local.

(Windows) Vous pouvez également déplacer le curseur sur l'étiquette du sélecteur et appuyer sur la touche <Ctrl> tout en déplaçant la roulette de la souris. **(UNIX)** Appuyez sur la touche <Meta>.



Le terminal de timeout situé dans le coin supérieur gauche de la structure Événement, représenté à gauche, spécifie la durée d'attente d'un événement, exprimée en millisecondes, qui doit s'écouler avant qu'une mise hors délai n'ait lieu. La valeur par défaut est -1, ce qui spécifie d'attendre indéfiniment qu'un événement se produise. Si vous câblez une valeur au terminal de timeout, vous devez fournir une condition Timeout.



Le nœud de données d'événement, représenté à gauche, se comporte de manière semblable à la fonction Désassembler par nom. Ce nœud est attaché au bord intérieur gauche de chaque condition d'événement. Ce nœud identifie les données que LabVIEW renvoie lorsqu'un événement se produit. Vous pouvez redimensionner ce nœud verticalement pour ajouter d'autres éléments de données et vous pouvez définir chaque élément de données dans le nœud pour accéder à n'importe quel élément de données d'événement. Le nœud fournit des éléments de données qui sont différents 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 seule condition pour gérer plusieurs événements, le nœud de données d'événement fournit uniquement les éléments de données d'événement qui sont communs à tous les événements configurés pour cette condition.



Le nœud de filtrage d'événements, représenté à gauche, est semblable au nœud de données d'événement. Ce nœud est attaché au bord intérieur droit des conditions d'événement de type filtre. Ce nœud identifie les sous-ensembles de données disponibles dans le nœud de données d'événement que la condition d'événement peut modifier. Il affiche des données différentes suivant l'événement pour lequel vous avez configuré cette condition. Par défaut, ces éléments correspondent aux éléments de données dans le nœud de données d'événement. Si vous ne câblez pas de valeur à un élément de données dans le nœud de filtrage d'événements, cet élément de données reste inchangé. Reportez-vous à la section [Événements de type filtre et notification](#) de ce chapitre pour obtenir de plus amples informations sur les événements de type filtre.



Pour accéder aux terminaux d'événement dynamique, représentés à gauche, cliquez avec le bouton droit sur la structure Événement et sélectionnez **Afficher les terminaux d'événement dynamique** dans le menu local. Ces terminaux sont utilisés uniquement pour l'enregistrement dynamique d'un événement. Reportez-vous aux sections [Enregistrement dynamique d'événements](#) et [Modification dynamique de l'enregistrement](#) de ce chapitre pour obtenir de plus amples informations sur l'utilisation de ces terminaux.



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.

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations complémentaires sur les valeurs par défaut pour les types de données.

Événements de type filtre et notification

Il existe deux types d'événements d'interface utilisateur : notification et filtre.

Les événements de type notification indiquent qu'une action utilisateur a déjà eu lieu, comme le changement de la valeur d'une commande. Utilisez les événements de type notification pour répondre à un événement après qu'il a eu lieu et que LabVIEW l'a traité. Vous pouvez configurer n'importe quel nombre de structures Événement pour répondre au même événement notification sur un objet spécifique. Lorsque l'événement a lieu, LabVIEW envoie une copie de l'événement à chaque structure Événement configurée pour gérer l'événement en parallèle.

Les événements de type filtre informent du fait que l'utilisateur a effectué une action avant que LabVIEW ne la traite, ce qui vous permet de personnaliser la façon dont le programme répond aux interactions avec l'interface utilisateur. Utilisez les événements de type filtre pour participer à la gestion de l'événement, en écrasant éventuellement le comportement par défaut de l'événement. Dans une condition de structure Événement pour un événement de type filtre, vous pouvez valider ou modifier les données d'événement avant que LabVIEW ne termine de les traiter ou vous pouvez entièrement détruire l'événement, empêchant ainsi la modification d'affecter le VI. Par exemple, vous pouvez configurer une structure Événement pour détruire l'événement "Face-avant fermée ?", empêchant ainsi l'utilisateur de fermer la face-avant du VI de façon interactive. Les événements de type filtre possèdent des noms qui se terminent par un point d'interrogation, comme "Face-avant fermée ?", afin de vous aider à les distinguer des événements de type notification. La plupart des événements de type filtre possèdent un événement notification du même nom qui leur est associé, mais sans le point d'interrogation, ce que LabVIEW génère après l'événement de type filtre si aucune condition d'événement n'a détruit l'événement.

Comme avec les événements de type notification, vous pouvez configurer n'importe quel nombre de structures Événement pour répondre au même événement filtre sur un objet spécifique. Toutefois, LabVIEW envoie les événements filtre de manière séquentielle à chaque structure Événement configurée pour cet événement. L'ordre dans lequel LabVIEW envoie l'événement à chaque structure Événement dépend de l'ordre dans lequel les événements ont été enregistrés. Reportez-vous à la section [Utilisation des événements dans LabVIEW](#) de ce chapitre pour obtenir de plus amples informations sur l'enregistrement d'un événement. Chaque structure Événement doit terminer sa condition pour cet événement avant que LabVIEW ne puisse notifier la structure Événement suivante. Si une

condition de structure Événement modifie des données d'événement, LabVIEW transmet les modifications aux structures Événement suivantes dans la chaîne. Si une structure Événement dans la chaîne détruit l'événement, LabVIEW ne transmet d'événement à aucune structure Événement restante dans la chaîne. LabVIEW finit de traiter l'action utilisateur qui avait déclenché l'événement uniquement après que toutes les structures Événement configurées ont traité l'événement sans le détruire.



Remarque National Instruments vous recommande d'utiliser des événements de type filtre uniquement lorsque vous souhaitez participer au traitement de l'action utilisateur, soit en détruisant l'événement, soit en modifiant les données d'événement. Si vous voulez simplement savoir si l'utilisateur a effectué une action particulière, utilisez les événements notification.

Les conditions de structure Événement qui gèrent les événements filtre ont un nœud de filtrage d'événements, comme décrit dans la section *Composantes de la structure Événement* section de ce chapitre. Vous pouvez modifier les données d'événement en câblant de nouvelles valeurs à ces terminaux. Si vous ne câblez pas d'élément de données, cet élément demeure inchangé. Vous pouvez détruire complètement un événement en câblant une valeur VRAI au terminal **Annuler** ?.



Remarque Une seule condition dans la structure Événement ne peut traiter à la fois un événement notification et un événement filtre. Une condition peut traiter plusieurs événements notification, mais ne peut traiter plusieurs événements filtre que si les éléments de données d'événement sont identiques pour tous les événements.

Utilisation des événements dans LabVIEW

LabVIEW peut générer un grand nombre d'événements différents. Pour éviter de générer des événements superflus, utilisez l'enregistrement d'événements afin de spécifier de quels événements vous souhaitez que LabVIEW vous avertisse. LabVIEW supporte deux modèles d'enregistrement d'événements : l'enregistrement statique et l'enregistrement dynamique.

L'enregistrement statique vous permet de spécifier quels événements de la face-avant d'un VI vous voulez traiter dans chaque condition de structure Événement sur le diagramme de ce VI. LabVIEW enregistre automatiquement ces événements lors de l'exécution du VI. Chaque événement est associé à une commande sur la face-avant du VI, à la totalité de la face-avant du VI ou à l'application LabVIEW. Vous ne pouvez pas configurer de manière statique une structure Événement pour qu'elle traite les événements de la face-avant d'un autre VI. La configuration est dite

statique car vous ne pouvez pas changer en cours d'exécution les événements qui sont traités par la structure Événement. Reportez-vous à la section [Enregistrement statique d'événements](#) de ce chapitre pour obtenir de plus amples informations sur l'utilisation de l'enregistrement statique.

L'enregistrement dynamique permet de transcender les limites de l'enregistrement statique en intégrant l'enregistrement d'événements au VI Serveur, ce qui vous permet d'utiliser les références d'application, de VI et de commande pour spécifier lors de l'exécution quels sont les objets pour lesquels vous voulez générer des événements. L'enregistrement dynamique procure une plus grande souplesse permettant de contrôler plus étroitement quels événements LabVIEW génère et à quel moment il les génère. Toutefois, l'enregistrement dynamique est plus complexe que l'enregistrement statique car il nécessite d'utiliser des références du VI Serveur avec les fonctions du diagramme pour procéder explicitement à l'enregistrement et au désenregistrement à des événements au lieu de traiter l'enregistrement automatiquement à l'aide des informations que vous avez configurées dans la structure Événement. Reportez-vous à la section [Enregistrement dynamique d'événements](#) de ce chapitre pour obtenir de plus amples informations sur l'utilisation d'enregistrements dynamiques.



Remarque En général, LabVIEW génère des événements d'interface utilisateur uniquement en résultat d'une interaction directe de l'utilisateur avec la face-avant active. LabVIEW ne génère pas d'événements, tels qu'un Changement de valeur, lorsque vous utilisez le VI Serveur, des variables globales, des variables locales, DataSocket et autres. Reportez-vous à la propriété Valeur (signalisation) dans l'*Aide LabVIEW* pour obtenir des informations complémentaires sur la manière de générer par programmation un événement Valeur changée. La plupart du temps, vous pouvez utiliser des événements générés par programmation au lieu de files d'attente et de notificateurs.

Les données sur l'événement fournies par un événement LabVIEW comprennent toujours un horodatage, une énumération indiquant quel événement s'est produit et une référence du VI Serveur à l'objet qui a déclenché l'événement. L'horodatage est un compteur en millisecondes que vous pouvez utiliser pour calculer le laps de temps écoulé entre deux événements ou pour déterminer dans quel ordre les événements ont eu lieu. La référence à l'objet qui a généré l'événement est de type strict et correspond à la classe VI Serveur de cet objet. Les événements sont groupés par classes selon le type d'objet qui a généré l'événement, comme par exemple Application, VI ou Commande. Si une condition unique gère plusieurs événements pour des objets de classes du VI Serveur différentes, le type de la référence est celui de la classe parente commune à tous les objets. 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. Reportez-vous au chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur l'utilisation des classes du VI Serveur.



Remarque Si vous enregistrez un même événement à la fois sur la classe VI et la classe Commande, LabVIEW génère l'événement du VI en premier. LabVIEW génère des événements de Commande pour les objets de type conteneur, comme les clusters, avant de générer des événements pour les objets qu'ils contiennent. Si la condition de structure Événement pour un événement de VI ou un événement de Commande sur un objet conteneur détruit l'événement, LabVIEW ne génère pas d'autres événements.

Chaque structure Événement et chaque nœud Enregistrement à des événements sur le diagramme possède une file d'attente que LabVIEW utilise pour stocker les événements. Lorsqu'un événement se produit, LabVIEW place une copie de l'événement dans chaque file d'attente enregistrée pour cet événement. La structure Événement gère tous événements dans sa file d'attente et éventuellement, ceux de tous les nœuds Enregistrement à des événements que vous avez câblés aux terminaux d'événements dynamiques de la structure Événement. LabVIEW utilise ces files d'attente pour s'assurer que tous les événements soient transmis de manière fiable à chaque structure Événement enregistrée et ce, dans l'ordre dans lequel ils ont eu lieu.

Par défaut, LabVIEW verrouille la face-avant contenant l'objet qui génère un événement jusqu'à ce que toutes les structures Événement aient fini de traiter l'événement. Tant que la face-avant est verrouillée, LabVIEW ne traite pas l'activité qui a lieu sur la face-avant, mais place ces interactions dans un buffer pour les traiter une fois que la face-avant est déverrouillée. Le verrouillage de la face-avant n'affecte pas certaines actions, comme déplacer la fenêtre, utiliser les barres de défilement et cliquer sur le bouton **Abandonner l'exécution**. Vous pouvez désactiver le verrouillage de la face-avant pour les événements notification en désélectionnant l'option dans la boîte de dialogue **Éditer les événements**. Le verrouillage de la face-avant doit être activé pour les événements de type filtre afin de s'assurer que l'état interne de LabVIEW n'est pas modifié avant qu'il ait eu le temps de terminer complètement le traitement de l'événement en cours.

LabVIEW peut générer des événements même lorsqu'aucune structure Événement n'attend de les gérer. Comme la structure Événement traite seulement un événement chaque fois qu'elle s'exécute, placez la structure

Événement dans une boucle While pour garantir qu'une structure Événement sera capable de gérer tous les événements qui ont lieu.



Mise en garde Si aucune structure Événement ne s'exécute pour traiter un événement et que le verrouillage de la face-avant est activé, l'interface utilisateur du VI ne répond plus. Dans ce cas-là, cliquez sur le bouton **Abandonner l'exécution** pour arrêter le VI. Vous pouvez désactiver le verrouillage de la face-avant en effectuant un clic droit sur la structure Événement et en décochant l'option **Verrouiller la face-avant jusqu'à ce que la condition d'événement finisse de s'exécuter** dans la boîte de dialogue **Éditer les événements**. Vous ne pouvez pas désactiver le verrouillage de la face-avant pour les événements de type filtre.

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

Enregistrement statique d'événements

L'enregistrement statique des événements est disponible uniquement pour les événements de l'interface utilisateur. Utilisez la boîte de dialogue **Éditer les événements** pour configurer une structure Événement afin qu'elle gère un événement enregistré de manière statique. Sélectionnez la source d'événement, qui peut être l'application, le VI ou une commande individuelle. Sélectionnez un événement spécifique que la source d'événement peut générer, comme Face-avant redimensionnée, Valeur changée et ainsi de suite. Éditez la condition pour traiter les données d'événement suivant les exigences de l'application. Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur la boîte de dialogue **Éditer les événements** et sur la manière d'enregistrer les événements de façon statique.

LabVIEW procède à l'enregistrement statique de façon automatique et transparente lorsque vous exécutez un VI qui contient une structure Événement. LabVIEW génère des événements pour un VI uniquement lorsque ce VI est en cours d'exécution ou lorsqu'un autre VI en cours d'exécution appelle ce VI en tant que sous-VI.

Lorsque vous exécutez un VI, LabVIEW définit ce VI de niveau principal et la hiérarchie des sous-VIs que le VI appelle sur son diagramme à un état d'exécution appelé réservé. Vous ne pouvez pas éditer un VI ou cliquer sur le bouton **Exécuter** tant que le VI est dans l'état réservé, du fait que le VI peut être appelé à tout moment comme sous-VI tant que son parent est en cours d'exécution. Quand LabVIEW définit un VI à l'état réservé, il enregistre automatiquement les événements que vous avez configurés de manière statique dans toutes les structures Événement du diagramme de ce

VI. Lorsque le VI de niveau principal termine son exécution, LabVIEW définit ce VI et sa hiérarchie de sous-VIs à l'état d'exécution inactive et annule automatiquement l'enregistrement des événements.

Reportez-vous à `exemples\general\uievents.llb` pour consulter des exemples d'utilisation d'enregistrements statiques d'événements.

Enregistrement dynamique d'événements

L'enregistrement dynamique d'événements vous laisse maître de décider quels événements seront générés par LabVIEW et à quel moment ils seront générés. Utilisez l'enregistrement dynamique si vous voulez que la génération d'événements n'ait lieu que lors d'une partie seulement de l'application ou si vous voulez changer la liste des VIs ou commandes qui génèrent des événements lorsque votre application est en cours d'exécution. Avec l'enregistrement dynamique, vous pouvez gérer des événements dans un sous-VI et non pas uniquement dans le VI dans lequel les événements sont générés.

La gestion d'événements enregistrés dynamiquement implique les quatre étapes principales suivantes :

1. Obtenir les références du VI Serveur aux objets pour lesquels vous voulez gérer les événements.
2. Enregistrer les objets aux événements en câblant les références du VI Serveur au nœud Enregistrement à des événements.
3. Entourez la structure Événement d'une boucle While pour traiter les événements sur les objets jusqu'à ce qu'une condition de fin d'exécution ait lieu.
4. Utilisez la fonction Désenregistrement à des événements pour arrêter la génération d'événements.

Pour enregistrer un objet à des événements de manière dynamique, obtenez d'abord une référence de VI Serveur à cet objet. Utilisez les fonctions Ouvrir une référence d'application et Ouvrir une référence de VI pour obtenir les références du VI et de l'application. Pour obtenir une référence de commande, utilisez un nœud de propriété afin d'interroger un VI sur ses commandes ou cliquez avec le bouton droit sur la commande et sélectionnez **Créer»Référence** dans le menu local pour créer une constante de référence à une commande. Reportez-vous à la section *Références d'application et de VI* du chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur l'utilisation des références du VI Serveur.

Utilisez la fonction Enregistrement à des événements pour un enregistrement dynamique à des événements. Vous pouvez redimensionner le nœud Enregistrement à des événements pour afficher une ou plusieurs entrées sources d'événements. Câblez une référence d'application, de VI ou de commande à chaque entrée source d'événement. Cliquez avec le bouton droit sur chaque entrée et sélectionnez l'événement que vous voulez enregistrer à partir du menu local **Événements**. Les événements parmi lesquels vous pouvez sélectionner dépendent du type de référence de VI Serveur que vous câblez à l'entrée source d'événement. Les événements dans le menu local **Événements** sont les mêmes que ceux qui apparaissent dans la boîte de dialogue **Éditer les événements** lorsque vous effectuez un enregistrement statique à des événements. Lorsque le nœud Enregistrement à des événements s'exécute, LabVIEW effectue un enregistrement pour l'événement spécifié sur l'objet qui est associé à chaque référence d'entrée source d'événement. Une fois que vous avez enregistré les événements, LabVIEW place les événements dans une file d'attente au fur et à mesure qu'ils se produisent. Les actions qui ont lieu avant que le nœud ne s'exécute ne génèrent pas d'événements à moins qu'un autre objet n'ait déjà été enregistré à ces événements.



Remarque Contrairement au nœud de propriété, le nœud Enregistrement à des événements ne nécessite pas que vous câblez l'entrée supérieure gauche. Utilisez cette entrée uniquement lorsque vous voulez modifier un enregistrement existant. Reportez-vous à la section [Modification dynamique de l'enregistrement](#) de ce chapitre pour obtenir de plus amples informations sur le ré-enregistrement d'un événement.

Les terminaux d'événements dynamiques, disponibles en cliquant avec le bouton droit sur la structure Événement et en sélectionnant **Afficher les terminaux d'événement dynamique** dans le menu local, se comportent de manière semblable aux registres à décalage. Le terminal gauche accepte un refnum d'enregistrement à des événements ou un cluster de refnums d'enregistrement à des événements. Si vous ne câblez pas le terminal intérieur droit, il contient les mêmes données que le terminal gauche. Vous pouvez toutefois câbler le refnum d'enregistrement à des événements ou le cluster de refnums d'enregistrement à des événements au terminal intérieur droit par l'intermédiaire d'un nœud Enregistrement à des événements et modifier l'enregistrement à des événements de manière dynamique. Reportez-vous à la section [Modification dynamique de l'enregistrement](#) de ce chapitre pour obtenir de plus amples informations sur l'utilisation des terminaux d'événements dynamiques.

La sortie du nœud Enregistrement à des événements est un refnum d'enregistrement à des événements, c'est-à-dire un type de données strict que LabVIEW utilise pour transmettre à la structure Événement des

informations sur l'enregistrement à des événements. Vous pouvez afficher les événements enregistrés dans la fenêtre d'**Aide contextuelle** en posant le curseur sur le refnum d'enregistrement à des événements. Une fois que vous avez configuré le nœud Enregistrement à des événements, câblez la sortie **refnum d'enregistrement à des événements** du nœud Enregistrement à des événements au terminal d'événement dynamique situé sur le côté gauche de la structure Événement et configurez celle-ci pour qu'elle gère les événements enregistrés. Évitez les branchements sur un fil de liaison de refnum d'enregistrement à des événements car les branchements permettent aux structures Événement multiples de retirer des événements d'une file d'attente, ce qui crée une situation de compétition risquant de donner lieu à un comportement imprévisible.

Utilisez la boîte de dialogue **Éditer les événements** pour configurer une structure Événement afin qu'elle gère un événement enregistré de manière dynamique. La section **Sources d'événement** de cette boîte de dialogue contient le sous-titre **Dynamique** sous lequel est listée chaque source d'événement enregistré de manière dynamique. Les noms des sources d'événements sont les mêmes que les références que vous avez câblées au nœud Enregistrement à des événements connecté à la structure Événement et sont répertoriés dans le même ordre. Sélectionnez la source de l'événement que vous désirez dans la liste **Dynamique**. Notez que le même événement que vous avez enregistré à l'aide du nœud Enregistrement à des événements apparaît en surbrillance dans la section **Événements**. Après que vous avez sélectionné l'événement, éditez la condition afin qu'elle gère les données sur l'événement conformément aux exigences de l'application.

Pour arrêter la génération d'événements, câblez le terminal d'événement dynamique sur le côté droit de la structure Événement à l'entrée **refnum d'enregistrement à des événements** de la fonction Désenregistrement à des événements, à l'extérieur de la boucle While qui contient la structure Événement. Lors de l'exécution de la fonction Désenregistrement à des événements, LabVIEW annule l'enregistrement de tous les événements spécifiés par le refnum d'enregistrement à des événements, détruit la file d'attente d'événements associée au refnum, ainsi que tous les événements qui restaient dans la file d'attente. Si vous n'annulez pas l'enregistrement d'événements et que l'utilisateur effectue des actions qui génèrent des événements après que la boucle While contenant la structure Événements a terminé son exécution, LabVIEW place indéfiniment les événements en file d'attente, ce qui risque de bloquer le VI si vous configurez les événements pour verrouiller la face-avant. Si cela se produit, LabVIEW détruit la file d'attente d'événements dès que le VI devient inactif.

LabVIEW annule aussi automatiquement l'enregistrement de tous les événements lorsque le VI de niveau principal termine son exécution. Cependant, National Instruments vous recommande d'annuler explicitement l'enregistrement des événements, particulièrement dans une longue application, afin d'économiser les ressources en mémoire.

Exemple d'événement dynamique

Le diagramme de la figure 9-1 démontre comment utiliser l'enregistrement d'événement dynamique en prenant comme exemple l'événement Entrée de la souris sur une commande chaîne.

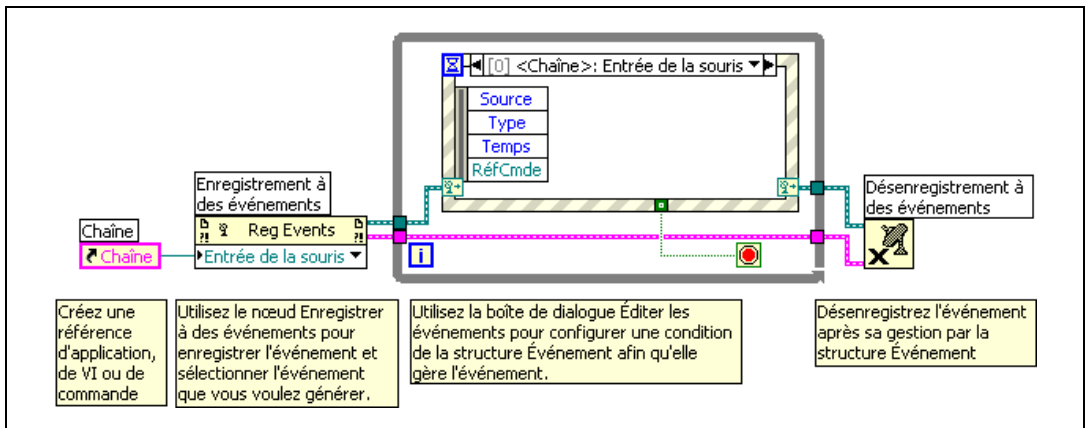


Figure 9-1. Enregistrement dynamique d'événements

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations complémentaires sur la manière d'enregistrer des événements de façon dynamique.

Reportez-vous à `exemples\general\dynamicevents.11b` pour consulter des exemples d'enregistrement dynamique d'événements.

Modification dynamique de l'enregistrement

Si vous enregistrez les événements de manière dynamique, vous pouvez modifier les informations d'enregistrement au moment de l'exécution pour changer les objets pour lesquels LabVIEW génère des événements. Câblez l'entrée supérieure gauche **refnum d'enregistrement à des événements** d'un nœud Enregistrer à des événements si vous voulez modifier l'enregistrement existant associé au refnum que vous avez câblé, plutôt que de créer un nouvel enregistrement. Lorsque vous câblez l'entrée **refnum d'enregistrement à des événements**, le nœud se redimensionne

automatiquement pour afficher les mêmes événements sur les mêmes types de références que vous aviez spécifiés dans le nœud Enregistrement à des événements qui avait initialement créé le refnum d'enregistrement. Vous ne pouvez pas redimensionner ou reconfigurer manuellement la fonction tant que l'entrée **refnum d'enregistrement à des événements** est câblée.

Si vous câblez une référence d'objet à l'entrée **source d'événement** d'un nœud Enregistrement à des événements et que vous avez aussi câblé l'entrée **refnum d'enregistrement à des événements**, le nœud remplace toute référence qui était précédemment enregistrée par l'entrée **source d'événement** correspondante du nœud Enregistrement à des événements d'origine. Vous pouvez câbler la constante Pas un refnum à une entrée **source d'événement** pour annuler l'enregistrement à un événement particulier. Si vous ne câblez pas d'entrée **source d'événement**, LabVIEW ne modifie pas l'enregistrement pour cet événement. Utilisez la fonction Désenregistrement à des événements si vous souhaitez annuler l'enregistrement pour tous les événements associés à un refnum d'enregistrement à des événements.

L'exemple de la figure 9-2 montre comment changer, en cours d'exécution et de manière dynamique, les objets pour lesquels LabVIEW génère des événements. Lorsque le diagramme suivant s'exécute, LabVIEW enregistre la référence Numérique et attend qu'un événement se produise sur la commande **Numérique** associée. Quand LabVIEW génère un événement Valeur changée pour la commande **Numérique**, la condition numérique Valeur changée exécute un nœud Enregistrement à des événements pour faire passer la commande numérique enregistrée pour l'événement Valeur changée de **Numérique** à **Numérique 2**. Si par la suite, l'utilisateur change la valeur de la commande **Numérique**, LabVIEW ne génère pas d'événement Valeur changée. Toutefois, des modifications de la commande **Numérique 2** génèrent des événements Valeur changée. Chaque fois que LabVIEW génère un événement Valeur changée pour la commande **Numérique 2**, le nœud Enregistrement à des événements s'exécute, mais reste sans effet car la commande **Numérique 2** est déjà enregistrée pour l'événement Valeur changée.

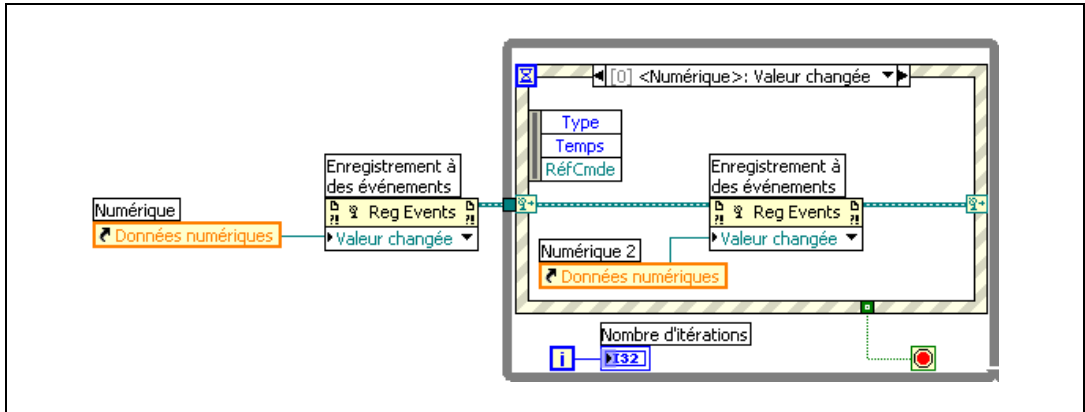


Figure 9-2. Modification dynamique de l'enregistrement à des événements



Remarque Vous ne pouvez pas modifier de manière dynamique des événements enregistrés de manière statique.

Événements utilisateur

Vous pouvez créer et nommer par programmation vos propres événements, appelés événements utilisateur, pour transmettre des données définies par l'utilisateur. Tout comme les files d'attente et les notificateurs, les événements utilisateur permettent à différentes parties d'une application de communiquer de manière asynchrone. Vous pouvez gérer à la fois l'interface utilisateur et les événements utilisateur générés par programmation dans la même structure Événement.

Création et enregistrement d'événements utilisateur

Pour définir un événement utilisateur, câblez un objet du diagramme, comme un terminal de face-avant ou une constante de diagramme, à la fonction Créer un événement utilisateur. Le type de données de l'objet définit le type de données de l'événement utilisateur. L'étiquette de l'objet devient le nom de l'événement utilisateur. Si le type de données est un cluster, le nom et le type de chaque champ du cluster définissent les données transmises par l'événement utilisateur. Si le type de données n'est pas un cluster, l'événement utilisateur comporte une seule valeur de ce type et l'étiquette de l'objet devient le nom de l'événement utilisateur et de l'unique élément de données.

La sortie **événement utilisateur de sortie** de la fonction Créer un événement utilisateur est un refnum de type strict qui contient le nom et le type de données de l'événement. Câblez la sortie **événement utilisateur de**

sortie de la fonction Créer un événement utilisateur à l'entrée **source d'événement** du nœud Enregistrement à des événements.

Traitez un événement utilisateur de la même manière que vous traitez un événement d'interface utilisateur enregistré de manière dynamique. Câblez la sortie **refnum d'enregistrement à des événements** du nœud Enregistrement à des événements au terminal d'événement dynamique situé sur le côté gauche de la structure Événement. Utilisez la boîte de dialogue **Éditer les événements** pour configurer une condition dans la structure Événement de façon à ce qu'elle gère l'événement. Le nom de l'événement utilisateur apparaît sous le sous-titre **Dynamique** de la section **Sources d'événement** de la boîte de dialogue.

Les éléments de données de l'événement utilisateur s'affichent dans le nœud de données d'événement sur le bord gauche de la structure Événement. Les événements utilisateur sont des événements de type notification qui peuvent partager la même condition d'événement d'une structure Événement que les événements d'interface utilisateur ou les autres événements utilisateur.

Vous pouvez câbler une combinaison d'événements utilisateur et d'événements d'interface utilisateur au nœud Enregistrement à des événements.

Génération d'événements utilisateur

Utilisez la fonction Générer un événement utilisateur pour transmettre l'événement utilisateur et les données qui lui sont associées à d'autres parties d'une application au moyen d'une structure Événement configurée pour gérer l'événement. La fonction Générer un événement utilisateur accepte un refnum d'événement utilisateur et une valeur pour les données sur l'événement. La valeur des données doit correspondre au type de données de l'événement utilisateur.

Si l'événement utilisateur n'est pas enregistré, la fonction Générer un événement utilisateur n'a aucun effet. Si l'événement utilisateur est enregistré, mais qu'aucune structure Événement ne l'attend, LabVIEW place l'événement utilisateur et ses données en file d'attente jusqu'à ce qu'une structure Événement s'exécute pour gérer l'événement. Vous pouvez effectuer plusieurs fois un enregistrement au même événement utilisateur en utilisant différents nœuds Enregistrement à des événements, auquel cas chaque file d'attente associée à un refnum d'enregistrement à des événements reçoit sa propre copie de l'événement utilisateur et des données qui lui sont associées chaque fois que la fonction Générer un événement utilisateur s'exécute.



Remarque Pour simuler une interaction utilisateur avec une face-avant, vous pouvez créer un événement utilisateur dont les éléments de données d'événement ont le même nom et le même type de données qu'un événement d'interface utilisateur existant. Par exemple, vous pouvez créer un événement utilisateur appelé `MyValChg` au moyen d'un cluster de deux champs booléens appelés `OldVal` et `NewVal`, qui sont les mêmes éléments de données d'événement que l'événement d'interface utilisateur Valeur changée associée à une commande booléenne. Vous pouvez partager la même condition de structure Événement pour l'événement utilisateur simulé `MyValChg` et un événement booléen réel Valeur changée. La structure Événement exécute la condition d'événement si un nœud Générer un événement utilisateur génère l'événement utilisateur ou si un utilisateur change la valeur de la commande.

Désenregistrement d'événements utilisateur

Désenregistrez les événements utilisateur lorsque vous n'en avez plus besoin. Détruisez en outre l'événement utilisateur en câblant le refnum d'événement utilisateur à l'entrée **événement utilisateur** de la fonction Détruire un événement utilisateur. Câblez la sortie **sortie d'erreur** de la fonction Désenregistrement à des événements à l'entrée **entrée d'erreur** de la fonction Détruire un événement utilisateur pour vous assurer que les fonctions s'exécutent dans l'ordre qui convient.

LabVIEW annule l'enregistrement de tous les événements et détruit automatiquement les événements utilisateur existants lorsque le VI de niveau principal termine son exécution. Cependant, National Instruments vous recommande d'annuler explicitement l'enregistrement des événements utilisateur et de les détruire, particulièrement dans une longue application, afin d'économiser les ressources en mémoire.

Exemple d'événement utilisateur

Le diagramme de la figure 9-3 démontre comment utiliser les événements utilisateur. Un cluster constante du diagramme spécifie le nom de l'événement utilisateur, `Événement utilisateur`, ainsi que le type de données de l'événement, c'est-à-dire une chaîne appelée `chaîne`. Le nœud Enregistrement à des événements effectue un enregistrement à l'événement utilisateur sur le refnum d'événement utilisateur. Une structure Événement dans une boucle While attend que l'événement se produise. Parallèlement à la boucle While, la fonction Générer un événement utilisateur envoie l'événement, ce qui force la condition Événement utilisateur à s'exécuter dans la structure Événement. Lorsque la boucle While termine son exécution, le VI annule l'enregistrement à l'événement et détruit l'événement utilisateur.

Construisez le VI de la figure 9-3 et utilisez ensuite le mode Exécution animée pour voir comment les données sur l'événement circulent entre les nœuds à travers le VI.

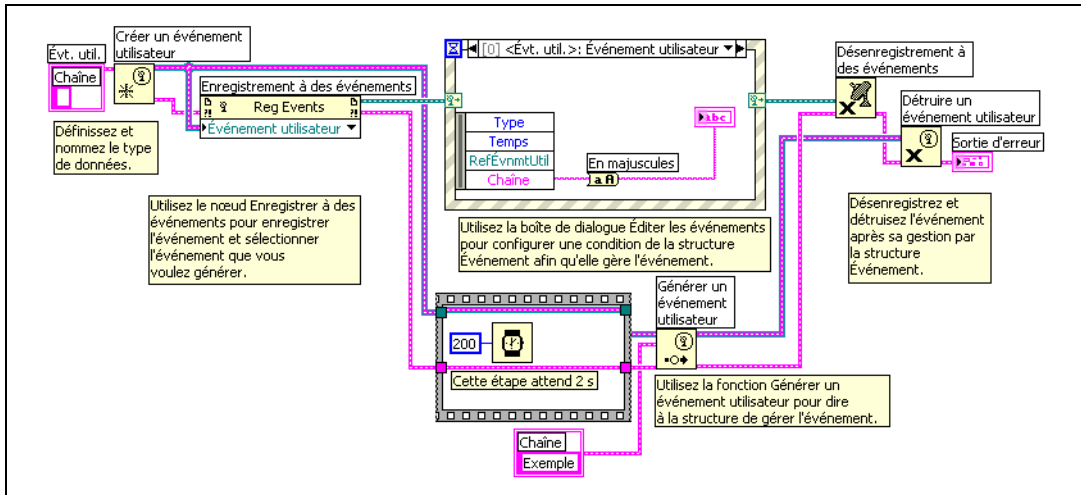


Figure 9-3. Génération d'événements utilisateur

Reportez-vous à `exemples\general\dynamicevents.llb` pour consulter des exemples d'enregistrement dynamique d'événements.

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 obtenir des informations complémentaires sur le groupement des données à l'aide de chaînes, de tableaux et de 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 les commandes et indicateurs de chaîne pour simuler des zones de saisie 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 obtenir de plus amples 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 10-1. La table présente également un exemple de message dans chaque type d'affichage.

Tableau 10-1. 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 affichables apparaissent généralement sous la forme d'un carré.	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\stypes\sd 'affichage.\n\\\sest\sune\sb arre\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 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 10-1 présente une table avec toutes ses composantes. Reportez-vous à la section [Tableaux](#) de ce chapitre pour obtenir de plus amples informations sur les tableaux.

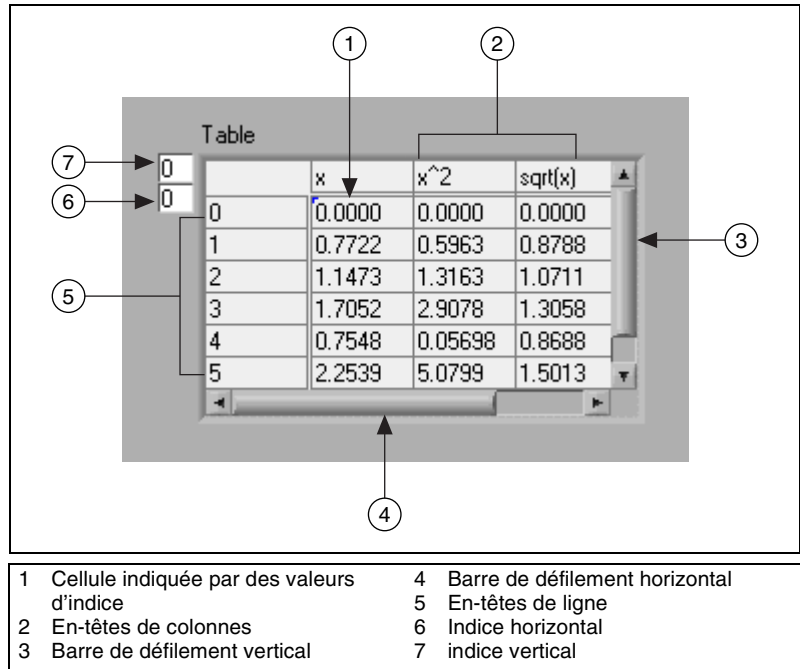


Figure 10-1. Composantes d'une table

Édition de chaînes par programmation

Utilisez les fonctions Chaîne pour modifier des chaînes de la façon suivante :

- 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 consulter des exemples d'utilisation des fonctions Chaîne servant à éditer des chaînes. Reportez-vous à la section *Memory and Speed Optimization* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière d'utiliser un minimum de mémoire lorsque vous éditez des chaînes par programmation.

Formatage de chaînes

Pour utiliser des données dans un autre VI, fonction ou application, vous devez souvent les convertir en chaîne de caractères, puis formater la chaîne de manière à ce que le VI, la fonction ou l'application puisse lire cette chaîne. Par exemple, Microsoft Excel requiert des chaînes de caractères contenant des délimiteurs, comme des tabulations, des virgules ou des espaces blancs. Excel utilise des délimiteurs pour séparer les nombres ou les mots en 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 par 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 pour réaliser 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 pour enregistrer des chaînes dans des fichiers texte et des fichiers de type tableur.

Spécificateurs de format

Le plus souvent, 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 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 10-2 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.

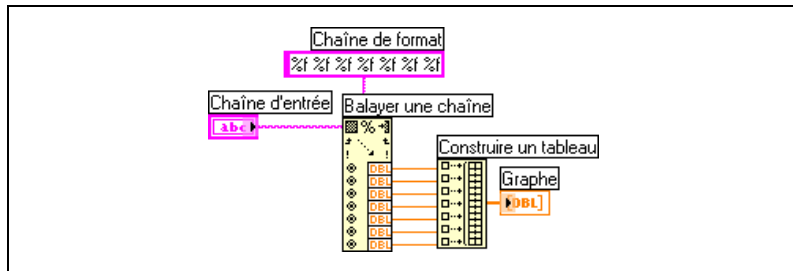


Figure 10-2. Conversion d'une chaîne en valeurs numériques

Conversion de données en données XML et vice versa

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 des 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 d'Allemagne* serait semblable à ceci :

```
<documentaire>
<Titre>Visiter les belles cathédrales
d'Allemagne</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 allemandes avec des
photographies en couleur, des coupes transversales à
l'échelle et l'historique de leur construction.</Résumé>
<Pages>224</Pages>
</documentaire>
```

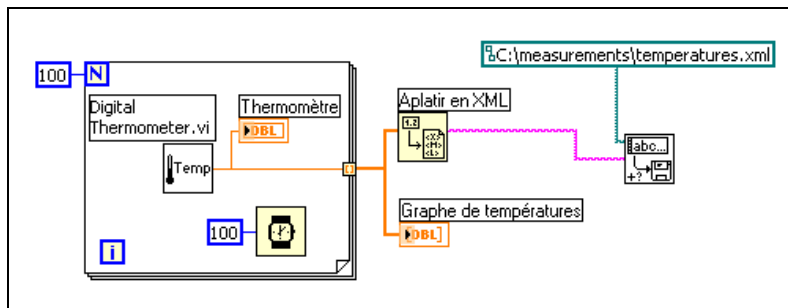
De manière similaire, vous pouvez classer des données LabVIEW par nom, valeur et type. Vous pouvez représenter en format XML une commande chaîne pour un nom d'utilisateur de la façon suivante :

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

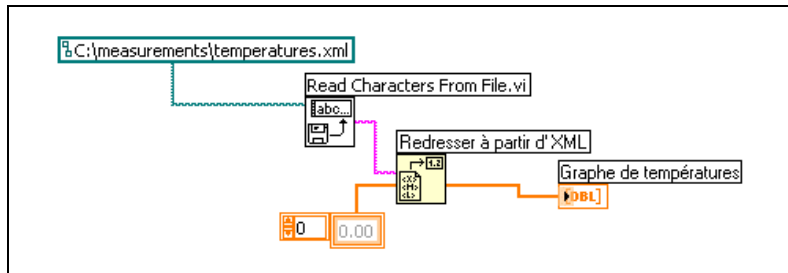
Utilisation de types de données basés sur le format XML

La conversion de données LabVIEW en XML formate les données de sorte que lorsque vous sauvegardez les données dans un fichier, vous pouvez identifier les valeurs, les noms et le type des données d'après les tags de description des données. 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 pour convertir un type de données 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 pour convertir des données au format XML en données LabVIEW. L'exemple suivant lit les 100 températures contenues dans le fichier `tempatures.xml` et trace le tableau de températures sur une graphe déroulant.



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

Reportez-vous à `exemples\file\XMLex.11b` pour consulter des exemples de conversion en format XML et à partir du format XML.

Schéma XML de LabVIEW

LabVIEW convertit des données en un schéma XML établi. Actuellement, vous ne pouvez pas créer de schémas personnalisés et vous ne pouvez pas contrôler la manière dont LabVIEW gère les tags de chaque partie de donnée. Par ailleurs, vous ne pouvez pas convertir la totalité de VIs ou de fonctions en XML.

Reportez-vous au répertoire `LabVIEW\help` pour consulter le schéma LabVIEW XML.

Groupement des données avec des tableaux et des clusters

Utilisez les commandes et les fonctions de tableaux et clusters pour grouper des données. Les tableaux regroupent les éléments de données du même type. Les clusters regroupent les éléments de données de types différents.

Tableaux

Un tableau est défini par des éléments et des 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](#) de ce chapitre pour obtenir de plus amples informations sur les types d'éléments pris en charge par les tableaux. Reportez-vous à la section [Clusters](#) de ce chapitre pour obtenir de plus amples 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 10-3.

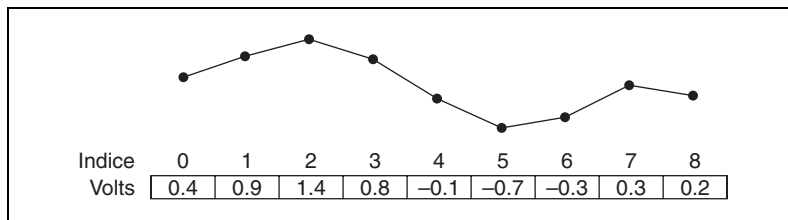


Figure 10-3. Signal dans un tableau de valeurs numériques

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 10-4.

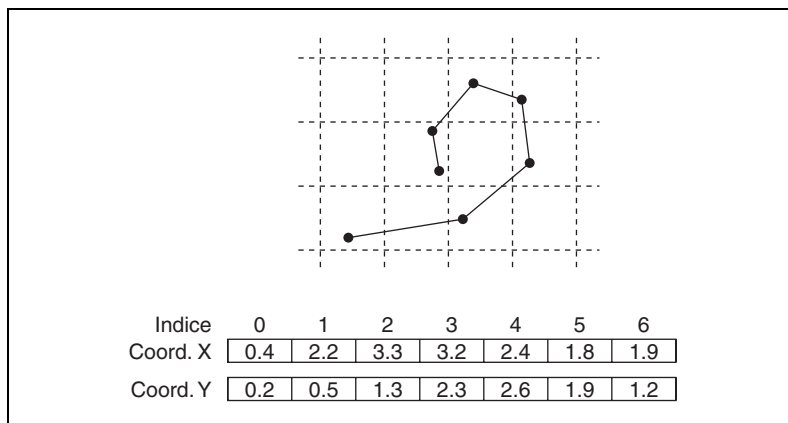


Figure 10-4. Graphe dans un tableau de points

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 10-5 présente un tableau 2D de 6 colonnes et 4 lignes, qui contient $6 \times 4 = 24$ éléments.

		Indice des colonnes					
		0	1	2	3	4	5
Indice des lignes	0						
	1						
	2						
	3						

Figure 10-5. 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 10-3 et 10-4 à deux dimensions en ajoutant une ligne au tableau. La figure 10-6 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.

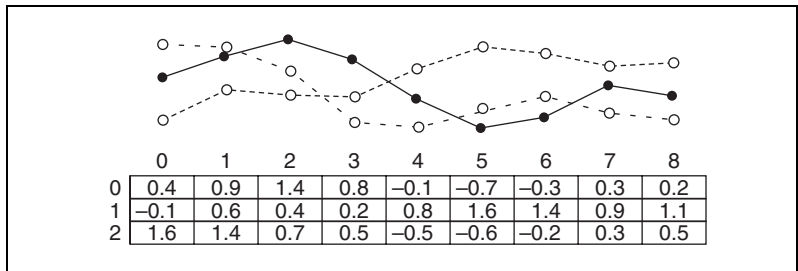


Figure 10-6. Signaux multiples dans un tableau 2D de valeurs numériques

Reportez-vous à `exemples\general\arrays.llb` pour consulter des exemples d'utilisation des tableaux. Reportez-vous à la section [Utilisation de boucles pour construire des tableaux](#) du chapitre 8, [Boucles et structures](#), pour obtenir des informations complémentaires sur la construction de tableaux.

Restrictions pour les tableaux

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

- Vous ne pouvez pas créer un tableau 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 commandes de face-avant secondaire.
- Vous ne pouvez pas créer un tableau de commandes onglet.
- Vous ne pouvez pas créer un tableau de commandes ActiveX.
- Vous ne pouvez pas créer un tableau de graphes déroulants.
- Vous ne pouvez pas créer un tableau de graphes XY multi-courbes.

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 10-7, et en faisant glisser dans ce tableau vierge un élément ou un objet de données, lequel peut être une commande ou un indicateur numérique, booléen, de chaîne, de chemin, de refnum ou de cluster.

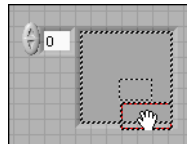


Figure 10-7. Tableau vierge

Le tableau vierge se redimensionne automatiquement pour accepter le nouvel objet, que ce soit une petite commande booléenne ou un grand graphe 3D. Pour créer un tableau multidimensionnel sur la face-avant, effectuez un clic droit sur l'affichage de l'indice et sélectionnez **Ajouter une dimension** dans le menu local. Vous pouvez aussi redimensionner l'affichage de l'indice jusqu'à ce que vous obteniez le nombre de dimensions souhaitées. Pour supprimer des dimensions une par une, cliquez avec le bouton droit sur l'affichage de l'indice et sélectionnez **Supprimer une dimension** dans le menu local. Vous pouvez aussi redimensionner l'affichage de l'indice pour supprimer les dimensions.

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 une constante tableau sur la palette de **Fonctions**, positionnez le tableau vierge sur la face-avant et placez dedans une constante chaîne, une constante numérique ou une constante cluster. 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 10-8, 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'affichage combiné à droite de l'affichage des lignes et colonnes présente la valeur à la position spécifiée. La figure 10-8 indique que la valeur de la ligne 6, colonne 13, est **66**.

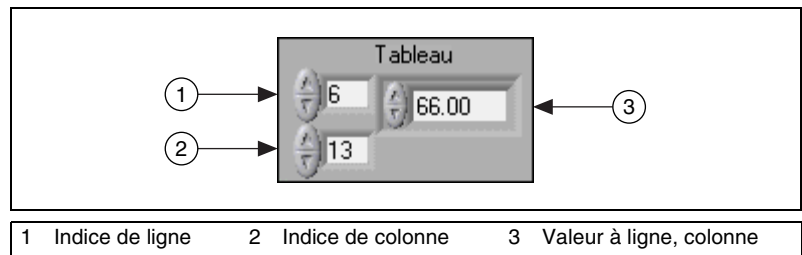


Figure 10-8. Commande tableau

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 redimensionner le tableau et afficher plusieurs lignes ou colonnes à la fois.

Fonctions de tableaux

Utilisez les fonctions Tableau pour créer et manipuler des tableaux, comme au moyen des 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 développez 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 10-9 utilise la fonction Indexer un tableau pour récupérer une ligne et un élément d'un tableau 2D.

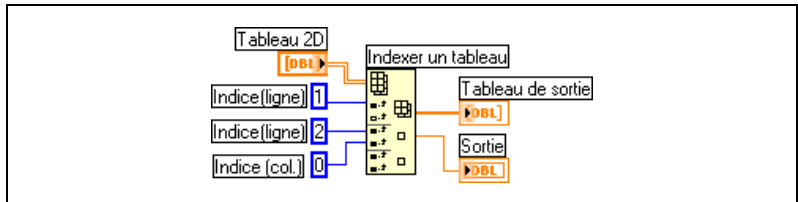


Figure 10-9. Indexation d'un tableau 2D

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**.

Reportez-vous à la section *Memory and Speed Optimization* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur la manière de minimiser l'utilisation de mémoire lorsque vous utilisez des fonctions Tableau dans une boucle.

Clusters

Les clusters regroupent des éléments de données de types différents, comme un assemblage de fils de liaison dans un câble de téléphone, par exemple, où chaque fil de liaison dans le câble représente un élément différent du cluster. Un cluster est semblable à un enregistrement ou 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 par programmation, 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 un terminal de type de données roses. Les clusters de valeurs numériques, appelés parfois des points, possèdent un modèle de fil de liaison et un terminal de type de données marron. Vous pouvez câbler les clusters numériques marron aux fonctions numériques, comme Additionner 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 s'affichent comme terminaux sur les fonctions Assembler et Désassembler du diagramme. Vous pouvez afficher 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 obtenir de plus amples informations sur la conversion numérique.

Utilisez les fonctions Cluster pour créer et manipuler des clusters. Par exemple, vous pouvez 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 de 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 obtenir des informations complémentaires 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 représentant la variable locale associée à l'objet apparaît sur le diagramme.



Vous pouvez aussi sélectionner une variable locale dans la palette de **Fonctions** et la placer sur le diagramme. Le nœud de variable locale, représenté à 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 un objet de la face-avant dans le menu local **Sélectionner un élément**. Le menu local répertorie tous les objets de la face-avant qui comportent des étiquettes liées.

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 obtenir de plus amples 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 des 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 une variable globale, représentée à gauche, dans la palette de **Fonctions** et placez cette variable sur le diagramme. Double-cliquez sur le nœud de variable globale pour afficher la face-avant du VI global. Placez 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 obtenir de plus amples 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 de face-avant. Un VI à variable globale avec plusieurs objets est plus efficace parce que vous pouvez grouper des variables apparentées. 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 obtenir de plus amples 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 menu local répertorie tous les objets de la face-avant qui comportent 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 un bord mince semblable à celui d'un indicateur.

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

Utilisation mesurée des variables globales et locales

Les variables globales et locales sont 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 de 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 de 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 obtenir de plus amples 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 ne la lise pour la première fois, la variable contient la valeur par défaut de l'objet de la face-avant associé.

Situations de compétition

Une condition de conflit apparaît lorsque deux parties du code ou davantage qui s'exécutent en parallèle changent la valeur de la même ressource partagée, typiquement une variable globale ou locale. La figure 11-1 présente un exemple de situation de compétition.

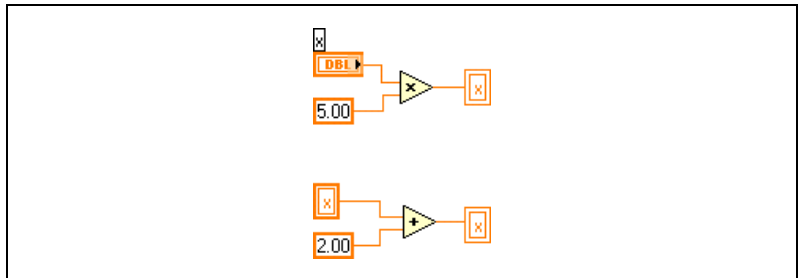


Figure 11-1. Situation de compétition

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. 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 artificielle des données* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur la dépendance des données.

Considérations sur la mémoire lors de l'utilisation de variables locales

Les variables locales effectuent des copies de 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. Si vous avez besoin d'enregistrer des données pendant l'exécution, envisagez d'utiliser un registre à décalage.

Considérations sur la mémoire lors de l'utilisation de variables globales

Lorsque vous lisez à partir d'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 enregistre et modifie 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, de la même manière qu'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 ceux qui sont déjà affichés pour créer un historique. 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 obtenir des informations complémentaires sur l'utilisation des graphes et des graphes déroulants.

Types de graphes et de graphes déroulants

Les graphes et les graphes déroulants 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 en tant qu'impulsions ou 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 `exemples\general\graphs.llb` pour consulter des exemples de graphes et de graphes déroulants.

Options des graphes et de graphes déroulants

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](#) de ce chapitre pour obtenir de plus amples 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](#) de ce chapitre pour obtenir de plus amples 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 supportent plusieurs échelles X et Y et tous les graphes déroulants supportent plusieurs échelles 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 du graphe ou du graphe déroulant et sélectionnez **Dupliquer l'échelle** dans le menu local pour ajouter plusieurs échelles 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 ou le graphe déroulant 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 *Formatage des axes* de ce chapitre pour obtenir de plus amples 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 X** : fait défiler les données dans le graphe ou le graphe déroulant. Utilisez la barre de défilement pour visualiser des données qui ne sont actuellement pas affichées par le graphe ou le graphe déroulant.

Personnalisation des graphes

Vous pouvez modifier le comportement des curseurs du graphe, les options d'échelle et les axes du graphe. La figure 12-1 présente les éléments d'un graphe.

Vous pouvez personnaliser les graphes numériques et les graphes d'intensité pour les adapter aux besoins de votre affichage de données ou pour afficher davantage 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.

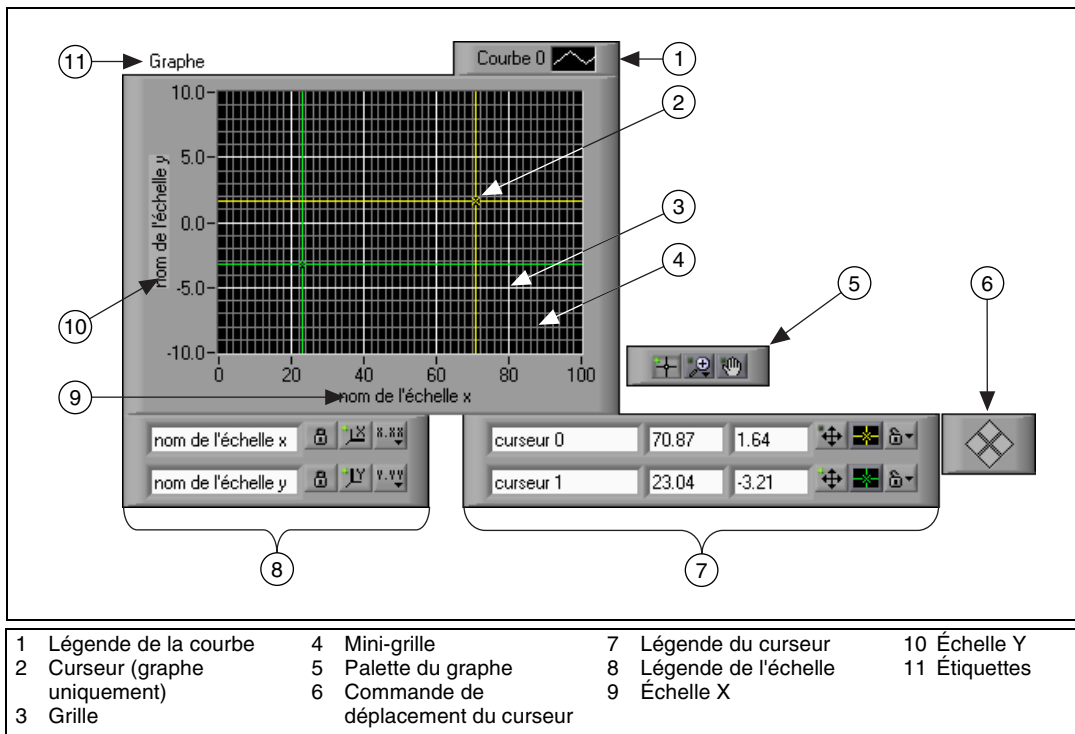


Figure 12-1. Éléments d'un graphe

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.

Courseurs 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.

Mise à l'échelle automatique

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 l'outil 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 axes

Utilisez l'onglet **Format et précision** des boîtes de dialogue **Propriétés du graphe** et **Propriétés du graphe déroulant** pour spécifier l'apparence de l'axe des x et de l'axe des y sur le graphe ou le graphe déroulant.

Par défaut, l'axe x est configuré pour utiliser une notation virgule flottante et son étiquette est `Temps` et l'axe y est configuré pour utiliser un formatage automatique et son étiquette est `Amplitude`. Cliquez avec le bouton droit sur le graphe ou le graphe déroulant et sélectionnez **Propriétés** pour afficher la boîte de dialogue **Propriétés du graphe** ou la boîte de dialogue **Propriétés du graphe déroulant** afin de configurer les axes du graphe ou du graphe déroulant.

Utilisez l'onglet **Format et précision** de la boîte de dialogue **Propriétés du graphe** ou **Propriétés du graphe déroulant** pour spécifier le format

numérique des axes du graphe ou du graphe déroulant. Sélectionnez l'onglet **Échelles** pour renommer les axes et pour formater l'apparence de l'échelle des axes. Par défaut, l'axe d'un graphe ou d'un graphe déroulant affiche jusqu'à six chiffres avant de basculer automatiquement à une notation exponentielle.

Cochez la case **Mode d'édition avancé** pour afficher les options de texte vous permettant d'entrer directement des chaînes de format. Entrez des chaînes de format afin de personnaliser l'apparence des échelles et la précision numérique des axes. Reportez-vous à la rubrique *Utilisation de chaînes de format* de l'*Aide LabVIEW* pour obtenir de plus amples informations sur les chaînes de formatage.

Formatage dynamique des graphes

Câblez une sortie type de données dynamique à un graphe pour formater automatiquement la légende de la courbe et l'horodatage de l'échelle des x du graphe. Par exemple, si vous configurez le VI Express Simuler un signal pour générer un signal sinusoïdal et pour utiliser un temps absolu, et que vous câblez la sortie du VI Express Simuler un signal à un graphe, la légende de la courbe du graphe met à jour automatiquement l'étiquette de la courbe pour qu'elle affiche `Signal sinusoïdal` et l'échelle des x affiche l'heure et la date lors de l'exécution du VI.

Cliquez avec le bouton droit sur le graphe et sélectionnez **Ignorer les attributs** dans le menu local pour ignorer l'étiquette de légende de courbe inclus dans le type de données dynamiques. Cliquez avec le bouton droit sur le graphe et sélectionnez **Ignorer l'horodatage** dans le menu local pour ignorer la configuration d'horodatage incluse dans le type de données dynamiques.



Remarque Si les données affichées dans le graphe contiennent un horodatage temps absolu, sélectionnez **Ignorer l'horodatage** pour ignorer l'horodatage. Reportez-vous au manuel *Initiation à LabVIEW* pour obtenir des informations complémentaires sur l'utilisation du type de données dynamique avec des graphes et pour obtenir plus de précisions sur les VIs Express. Reportez-vous à la section *Type de données dynamiques* du chapitre 5, *Construction du diagramme*, pour obtenir des informations complémentaires sur l'utilisation du type de données dynamique.

Utilisation du rafraîchissement progressif

Lorsque LabVIEW met à jour un objet alors que le rafraîchissement progressif est désactivé, il efface le contenu de l'objet et trace la nouvelle valeur, ce qui produit un scintillement visible. L'utilisation du

rafraîchissement progressif permet d'éviter le scintillement causé par le tracé et la suppression. 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 ralentir 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 numériques et les graphes déroulant d'intensité pour les adapter aux besoins de votre affichage de données ou pour afficher davantage 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 X** dans le menu local pour afficher une barre de défilement.

Mode de mise à jour du graphe déroulant

Les graphes déroulants utilisent trois modes d'affichage 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**.

- **Graphe déroulant** : affiche des données en cours d'exécution défilant en continu de gauche à droite dans le graphe déroulant, en plaçant les anciennes données sur la gauche et les nouvelles sur la droite. Un graphe déroulant est semblable à un enregistreur papier de graphe déroulant.
- **Oscillographe** : présente un élément de donnée, comme une impulsion ou une onde, défilant dans le graphe déroulant de gauche à droite. Le graphe déroulant représente chaque nouvelle valeur à droite de la valeur précédente. Lorsque la courbe atteint le bord droit de la zone de tracé, LabVIEW efface le tracé et recommence à tracer à partir du bord gauche. L'affichage du tracé d'un graphe oscillographe est semblable à la manière dont un oscilloscope effectue un tracé.
- **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. LabVIEW n'efface pas le tracé dans un graphe à balayage quand le tracé atteint le bord droit de la zone de tracé. Un graphe à balayage est similaire à un affichage d'électrocardiogramme.

Courbes superposées comparées aux 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 12-2 présente des exemples de courbes superposées et empilées.

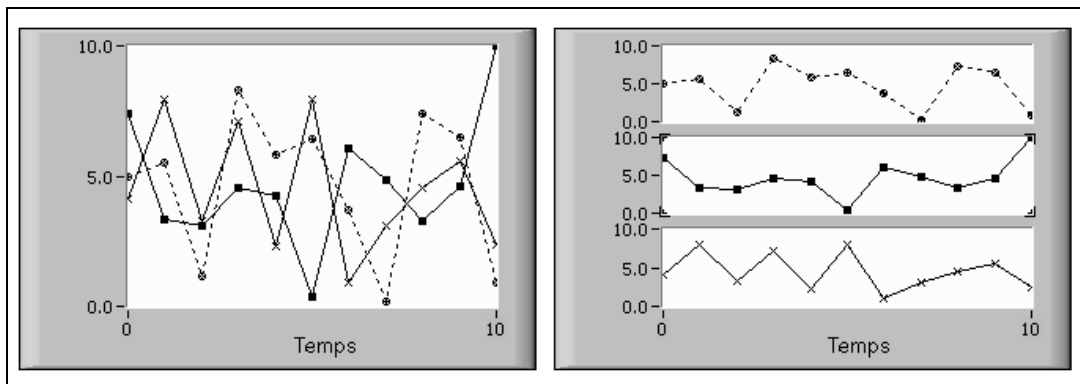


Figure 12-2. Graphes déroulants avec des courbes superposées et empilées

Cliquez avec le bouton droit sur le graphe déroulant et sélectionnez **Courbes empilées** dans le menu local pour afficher les courbes du graphe déroulants sous forme de multiples échelles verticales. Cliquez avec le bouton droit sur le graphe déroulant et sélectionnez **Courbes superposées** pour afficher les courbes du graphe déroulants sous forme d'une seule échelle verticale.

Reportez-vous au VI Charts de la bibliothèque `exemples\general\graphs\charts.llb` pour consulter 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 12-3 présente des exemples de graphes et de graphes XY.

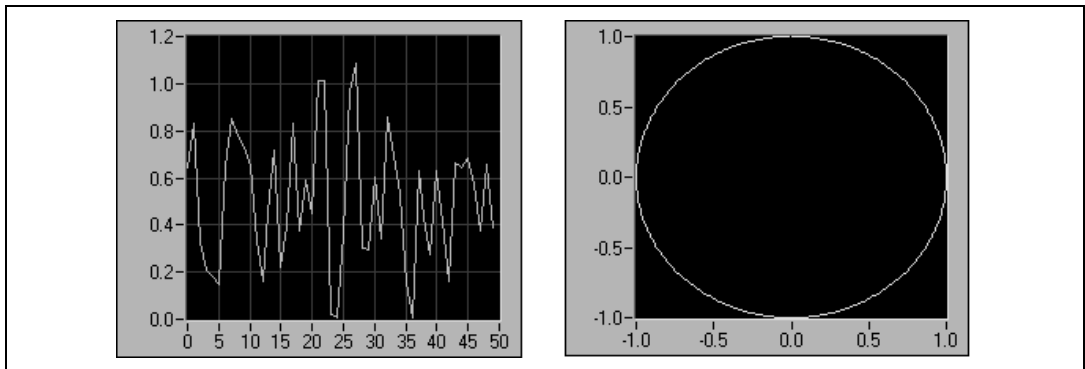


Figure 12-3. Graphes et graphes XY

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 waveforms 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.llb` pour consulter 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 renvoie les données sous forme de tableaux 2D, avec chaque voie stockée comme colonne séparée. Reportez-vous au graphe multi-courbes (Y) Multi Plot 1 du VI Waveform Graph de la bibliothèque `examples\general\graphs\gengraph.llb` pour consulter 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.llb` pour consulter 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 12-4 présente le tableau de cluster contenant les données y .



Figure 12-4. 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 consulter 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 consulter 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 consulter 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 consulter 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 consulter 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 consulter des exemples de graphes déroulants.

Si vous transmettez aux graphes déroulants une seule valeur ou plusieurs valeurs à la fois, LabVIEW interprète les données comme points sur le graphe déroulant et incrémente l'indice x de un, en partant de $x = 0$. Le graphe déroulant traite ces entrées comme nouvelles données pour une courbe unique. Si vous transmettez le type de données waveform aux graphes déroulants, l'indice x se conforme au format horaire spécifié.

La fréquence à laquelle vous envoyez des données au graphe déroulant détermine la fréquence de rafraîchissement du tracé dans le graphe déroulant.

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 courbes que vous souhaitez afficher avant l'exécution ou si vous désirez transmettre plusieurs points pour différentes courbes au cours d'une seule mise à jour, câblez un tableau 2D de données numériques ou waveform 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 et des graphes déroulants d'intensité pour afficher des données contenant des motifs, comme par exemple une carte de température ou une carte topographique, dans laquelle 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 12-5 présente le concept de l'opération pour le graphe déroulant d'intensité.

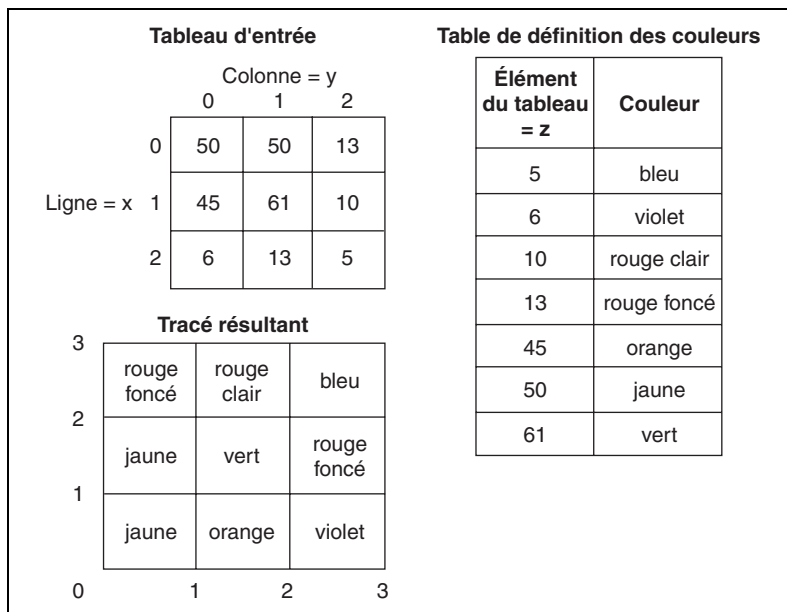


Figure 12-5. Carte de couleurs du graphe déroulant d'intensité

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.

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 [Modes de mise à jour du graphe déroulant](#) de ce chapitre pour obtenir de plus amples informations sur ces graphes déroulants.

Reportez-vous à la bibliothèque `examples\general\graphs\intgraph.lib` pour consulter 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 de plus amples 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 mappées 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 entre les valeurs et les 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 numérique pour afficher des données numériques, notamment lorsque vous utilisez des diagrammes temporels ou des analyseurs logiques. Référez-vous au *LabVIEW Measurements Manual* pour obtenir des informations complémentaires sur l'acquisition de données numériques.

Les graphes numériques acceptent le type de données waveform numériques, le type de données numériques et un tableau de ces type de données comme entrée. Par défaut, le graphe numérique condense les bus numériques de sorte qu'il puisse tracer les données numériques sur une seule courbe. Si vous câblez un tableau de données numériques, le graphe numérique trace chaque élément du tableau comme courbe distincte dans l'ordre du tableau.

Le graphe numérique de la figure 12-6 trace les données numériques sur une seule courbe. Le VI convertit les nombres du tableau **Nombres** en données numériques et affiche les représentations binaires des nombres dans l'indicateur de données numériques **Représentations binaires**. Dans le graphe numérique, le chiffre 0 apparaît sans ligne au-dessus pour indiquer que toutes les valeurs binaires sont des zéros. Le chiffre 255 apparaît sans ligne au-dessous pour indiquer que toutes les valeurs binaires sont des 1.

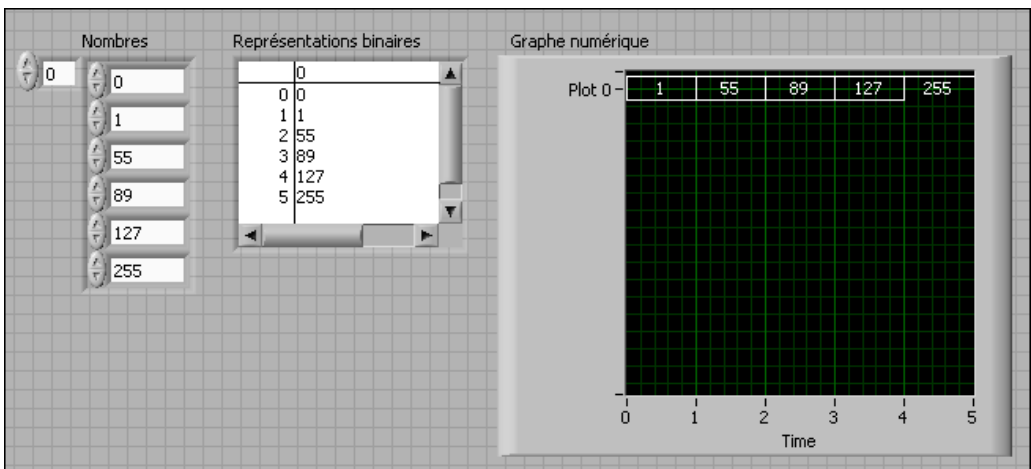


Figure 12-6. Tracé de données numériques sur une seule courbe

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur la configuration des courbes d'un graphe numérique.

Pour tracer chaque échantillon de données numériques, cliquez avec le bouton droit sur l'axe y et sélectionnez **Détailler les bus numériques** dans le menu local. Chaque courbe représente un échantillon différent.

Le graphe numérique de la figure 12-7 affiche les six chiffres de la commande de tableau numérique **Nombres**.

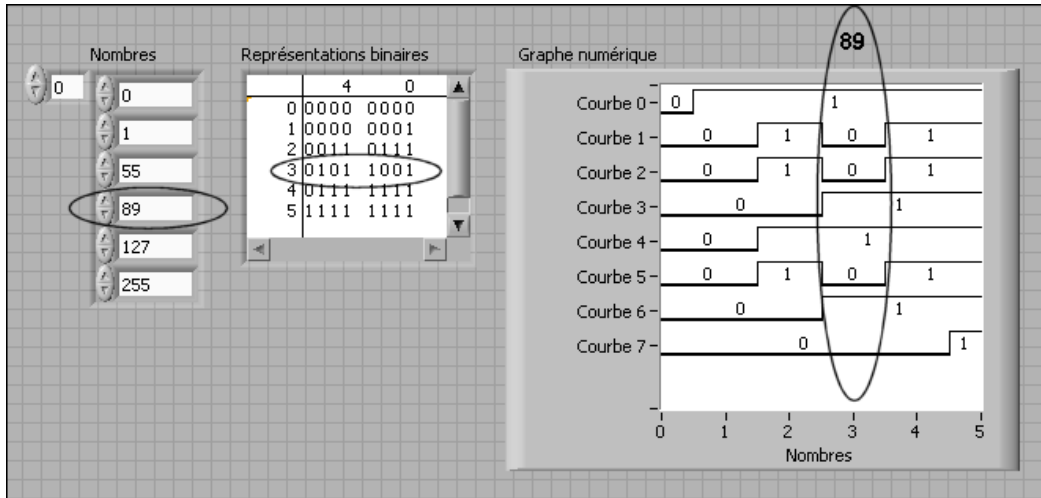


Figure 12-7. Représentation graphique d'entiers sous forme numérique

La commande numérique **Représentations binaires** affiche les représentations binaires des nombres. Chaque colonne du tableau représente un bit. Par exemple, le nombre 89 nécessite 7 bits de mémoire (le 0 dans la colonne 7 indique un bit inutilisé). Point 3 sur le graphe numérique trace les 7 bits nécessaires à la représentation du nombre 89 et la valeur zéro pour représenter le huitième bit inutilisé sur la courbe 7.

Le VI de la figure 12-8 convertit un tableau de nombres en données numériques et utilise la fonction Construire une waveform pour assembler l'heure de début, le delta t et les nombres entrés dans une commande de données numériques, et pour afficher les données numériques.

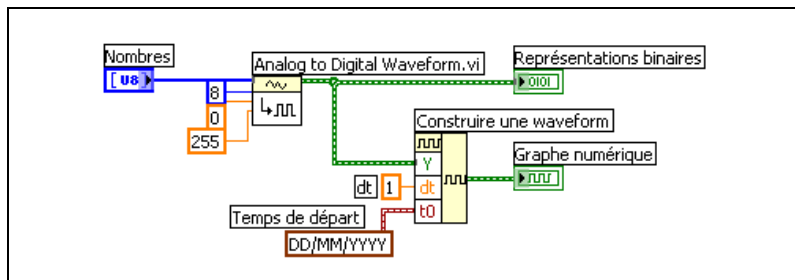


Figure 12-8. Conversion d'un tableau de nombres en données numériques

Reportez-vous aux sections *Commande de waveform numérique* et *Commande de données numériques* du chapitre 4, *Construction de la face-avant*, pour obtenir de plus amples informations sur la commande de

données numériques, le type de données waveform numériques et la conversion de données en données numériques.

Masquage des données

Le VI de la figure 12-7 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 la même courbe.

Graphes 3D



Remarque Les commandes de graphes 3D ne sont disponibles que sous Windows et dans les 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 afficher 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 :

- **Surface 3D** : trace une surface dans l'espace 3D. Lorsque vous placez cette commande sur la face-avant, LabVIEW la connecte à un sous-VI qui reçoit les données représentant la surface.
- **Graphe paramétrique 3D** : trace une surface complexe dans l'espace 3D. Lorsque vous placez cette commande sur la face-avant, LabVIEW la connecte à un sous-VI qui reçoit les données représentant la surface.
- **Courbe 3D** : trace une ligne dans l'espace 3D. Lorsque vous placez cette commande sur la face-avant, LabVIEW la connecte à un sous-VI qui reçoit les données représentant la ligne.

Utilisez les graphes 3D en conjonction avec les VIs 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 afficher 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 Propriétés des graphes 3D modifient leur comportement lors de l'exécution, notamment 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 transmet les données, l'heure de début et le Δt d'un signal. Vous pouvez créer des waveforms à l'aide de la fonction Construire une Waveform. Un grand nombre des VIs et des fonctions que vous utilisez pour acquérir et analyser les signaux acceptent et renvoient 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, *Creating a Typical Measurement Application*, du *LabVIEW Measurements Manual* pour obtenir de plus amples informations sur l'utilisation du type de données waveform.

Type de données waveform numérique

Le type de données waveform numérique comprend l'heure de début, le delta x, les données et les attributs d'une waveform numérique. Vous pouvez utiliser la fonction Construire une Waveform pour créer des waveforms numériques. Lorsque vous câblez un type de données waveform numérique à un graphe numérique, celui-ci trace automatiquement une waveform basée sur les informations de synchronisation et les données de la waveform numérique. Câblez le type de données waveform numérique à un indicateur de données numériques pour afficher les échantillons et les signaux d'une waveform numérique. Reportez-vous au chapitre 5, *Typical Measurement Applications*, du *LabVIEW Measurements Manual* pour obtenir de plus amples informations sur l'utilisation du type de données waveform numérique.

VIs Graphisme et son

Utilisez les VIs Graphisme et son pour afficher ou modifier les graphiques et les sons dans les VIs.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations complémentaires sur l'utilisation de graphiques et de sons dans les VIs.

Utilisation de l'indicateur image

L'indicateur image est un indicateur polyvalent servant à afficher des images contenant des lignes, des cercles, du texte et d'autres types de formes 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. Utilisez l'indicateur image et les VIs Graphisme au lieu d'une application de graphiques pour créer, modifier et afficher des graphiques dans LabVIEW.

L'indicateur image dispose d'un système de coordonnées basé sur les pixels, dans lequel l'origine (0) correspond au pixel situé à l'angle supérieur gauche de la commande. La composante horizontale (x) d'une coordonnée croît vers la droite, tandis que la coordonnée verticale (y) 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. Vous pouvez aussi afficher les barres de défilement verticale et horizontale pour un indicateur image afin que vous puissiez voir les pixels qui ne se trouvent pas dans la zone d'affichage de l'indicateur. Cliquez avec le bouton droit sur l'indicateur et sélectionnez **Éléments visibles» Barre de défilement** dans le menu local pour afficher les barres de défilement de l'indicateur.

Vous pouvez utiliser les propriétés du VI Serveur dans la classe Image pour modifier l'indicateur image par programmation, comme par exemple pour changer la taille de l'indicateur image ou sa zone d'affichage.

Reportez-vous au chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur l'utilisation des propriétés du VI Serveur.



Lorsque vous placez un indicateur image sur la face-avant, il apparaît sous la forme d'une zone rectangulaire vierge et un terminal correspondant, représenté à gauche, apparaît sur le diagramme.

Pour afficher une image dans un indicateur image, vous devez écrire une image dans l'indicateur par programmation. Vous ne pouvez pas copier une image dans le presse-papiers pour la coller ensuite dans l'indicateur image. Vous pouvez utiliser les VIs Fonctions sur les images pour spécifier un jeu d'instructions pour le tracé. Chaque VI correspond à un ensemble d'entrées décrivant une instruction graphique. En fonction de ces entrées, le VI crée une description compacte de ces spécifications que vous transmettez à l'indicateur image pour l'affichage. Reportez-vous à la section *VIs de fonctions sur les images* de ce chapitre pour obtenir des informations complémentaires sur l'utilisation des Fonctions sur les images pour afficher une image dans un indicateur image.

VIs de tracés d'images

Utilisez les VIs de tracés d'images pour créer des types de graphes courants à l'aide de l'indicateur image. Ces graphes comprennent un graphe polaire, un graphe de waveforms, un graphe XY, une abaque de Smith et une échelle de graphe.

Les VIs de tracés d'images utilisent 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. Par exemple, vous pouvez utiliser le VI Tracer une waveform pour créer un tracé présentant des fonctionnalités légèrement différentes de celles des graphes intégrés.

Utilisation du VI Graphe polaire comme sous-VI

Utilisez le VI Graphe polaire pour tracer 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 Graphe polaire 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 de la bibliothèque `examples\picture\demos.llb` et placez-la sur la face-avant.

Utilisation des VIs Tracer une waveform et Tracer XY comme sous-VIs

Utilisez le VI Tracer une waveform, qui simule le comportement du graphe intégré de LabVIEW, pour tracer des waveforms de styles variés, y compris des points, des points connectés et des barres. 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 Tracer une waveform réunit de nombreuses fonctionnalités en un seul VI. 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 à partir du VI Waveform and XY Plots de la bibliothèque `examples\picture\demos.llb` et placez-la sur la face-avant.

Le VI Tracer XY et le VI Tracer plusieurs XY sont semblables au VI Tracer une waveform. Vous utilisez des commandes différentes pour spécifier l'apparence du tracé parce que les VIs Tracer XY possèdent trois styles de courbe supplémentaires : 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 maximales et minimales de y pour cette valeur de x .

Utilisation des VIs Abaque de Smith comme sous-VIs

Utilisez les VIs Abaque de Smith 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 Abaque de Smith 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 Abaque de Smith comme aide visuelle pour concorder avec l'impédance et pour calculer le coefficient de réflexion d'une ligne de transmission.

Les VIs Abaque de Smith réunissent de nombreuses fonctionnalités en un seul VI. 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 à partir des exemples de VIs Abaque de Smith de la bibliothèque `examples\picture\demos.llb` et placez-la sur la 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 l'abaque de Smith, utilisez le VI Normaliser une abaque de Smith pour normaliser les données. Vous pouvez transmettre les données que vous normalisez avec le VI Normaliser une abaque de Smith directement au VI Abaque de Smith. Vous mettez généralement à l'échelle les données de l'abaque de Smith par rapport à l'impédance caractéristique (Z_0) du système.

VIs de fonctions sur les images

Utilisez les VIs Fonctions sur les images pour tracer 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 sont des tableaux 2D de couleurs, où chaque valeur correspond à une couleur ou à un indice dans un tableau de valeurs de couleurs RVB, selon la profondeur des couleurs.

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 utilisez les VIs Fonctions sur les images, l'image garde en mémoire la position du crayon graphique. Pour la plupart des Fonctions sur les images, vous devez spécifier des coordonnées absolues, c'est-à-dire relatives à l'origine (0, 0). Avec le VI Tracer une ligne et le VI Déplacer le crayon, vous pouvez spécifier des coordonnées absolues ou relatives. Des coordonnées relatives se réfèrent à la position actuelle du crayon. Vous pouvez utiliser le VI Déplacer le crayon pour changer la position du crayon sans dessiner. Seuls les VIs Tracer un point, Déplacer le crayon, Tracer une ligne et Tracer plusieurs lignes changent la position du crayon.

La seconde rangée de la palette **Fonctions sur les images** contient des VIs que vous utilisez pour tracer une forme. Chacun de ces VIs trace une forme dans la zone rectangulaire d'une image. Vous spécifiez un rectangle avec un cluster de quatre valeurs qui représentent les pixels gauche, haut, droit et bas.

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 Obtenir rectangle du texte ne trace pas de texte. Vous l'utilisez pour calculer la taille du rectangle de délimitation d'une chaîne.

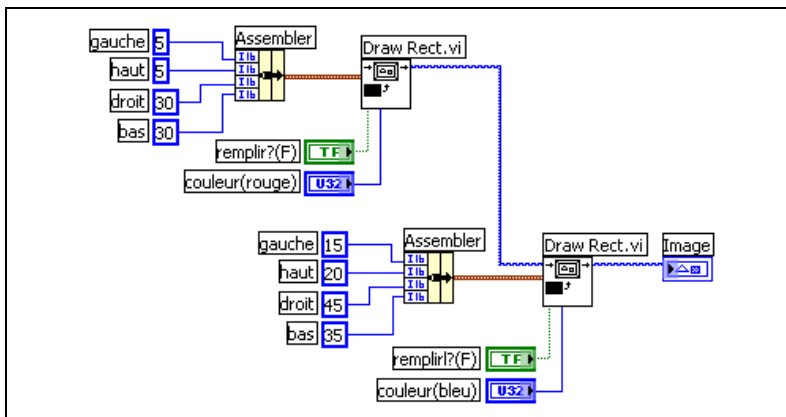
La quatrième rangée de la palette **Fonctions sur les images** contient des VIs qui servent à tracer des tables de pixels aplaties ou non aplaties dans une image, à appliquer un masque à une image, à obtenir une sous-image source et à convertir un type de données image en un cluster de données image aplaties.

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. La dernière rangée de cette palette contient aussi des VIs qui servent à convertir les valeurs rouge, vert et bleu dans la couleur RVB correspondante et à convertir une couleur en ses composantes respectives rouges, vertes et bleues.

Vous ne pouvez câbler les images créées avec les VIs Fonctions sur les images qu'à un indicateur image ou à l'entrée **image** d'un VI Fonctions sur les images. LabVIEW trace l'image lorsqu'il met à jour l'indicateur image sur une face-avant ouverte.

Chaque VI Fonctions sur les images enchaîne ses instructions graphiques aux instructions graphiques câblées à l'entrée **image** et renvoie les instructions graphiques concaténées dans la sortie **nouvelle image**.

Le diagramme suivant utilise le VI Tracer un rectangle pour tracer deux rectangles qui se chevauchent.



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. La façon la plus facile de spécifier une couleur est d'utiliser une constante boîte de couleurs 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 les valeurs bleue de chaque éléments changent et sont supérieures aux valeurs rouge et vert. 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 pour lire et écrire des données de plusieurs formats de fichiers graphiques standard, y compris les fichiers bitmap (.bmp), les fichiers Portable Network Graphics (.png) et les fichiers Joint Photographic Experts Group (.jpg). Vous pouvez utiliser ces VIs pour accomplir les tâches suivantes :

- Lire des données d'un fichier graphique pour les afficher dans un indicateur de type image.
- Lire des données pour manipuler des images.
- Écrire une image pour l'afficher dans d'autres applications.

Les données bitmap correspondent à des tableaux 2D dans lesquels 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 couleurs vraies, 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 le 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 directement ces données aplaties à l'aide du VI Tracer une table de pixels aplatie.

Pour manipuler les données sous forme de tableau 2D, convertissez-les au format approprié à l'aide du VI Redresser la table de pixels et du VI Aplatir la table de pixels.

VIs Son

Utilisez les VIs Son pour intégrer des fichiers et des fonctions audio à vos VIs. Vous pouvez utiliser ces VIs pour accomplir les tâches suivantes :

- Créer des VIs qui lisent 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 audio d'autres VIs Son. Vous pouvez contrôler le volume sonore dans le périphérique, lire ou mettre en pause le fichier audio ou le supprimer de votre système.

E/S sur fichiers

Les opérations d'E/S sur fichiers transmettent des données en provenance et en direction de fichiers. Utilisez les VIs et les fonctions d'E/S sur fichiers pour gérer tous les aspects d'E/S sur fichiers, notamment ceux-ci :

- 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 des informations complémentaires 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 de plus amples informations 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 auront 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 pour effectuer des lectures ou des écritures et si la vitesse et l'espace disque 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 au format texte, utilisez les 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 un octet d'espace, 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 écrivez les données dans le fichier texte. Le problème de la perte de précision ne se pose pas avec les fichiers binaires.

Reportez-vous à `examples\file\smp1file.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

Le stockage de données binaires, comme par exemple un entier, utilise un nombre fixe d'octets 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 des fichiers binaires pour enregistrer des données numériques et pour accéder à des nombres spécifiques d'un fichier ou accéder aléatoirement à des nombres 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 dans des fichiers binaires, mais cela n'est pas courant.

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 obtenir de plus amples 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 sous forme de séquence de caractères ou d'octets.

Reportez-vous aux VIs Read Binary File et Write Binary File dans le répertoire `examples\file\smpfile.llb` pour obtenir des exemples de lecture et d'écriture de tableaux de valeurs à virgule flottante double précision en provenance et en direction d'un fichier binaire, respectivement.

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.

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](#) de ce chapitre pour obtenir des informations complémentaires 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 `examples\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 pour effectuer des opérations d'E/S courantes, comme l'écriture ou la lecture des types de données suivants :

- Caractères en provenance ou en direction de fichiers texte
- Lignes à partir de fichiers texte.
- Tableaux 1D ou 2D de numériques simple précision en provenance ou en direction de fichiers texte de type tableur.
- Tableaux 1D ou 2D de numériques simple précision ou d'entiers signés 16 bits en provenance ou en direction 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 chaque fois qu'ils s'exécutent. Reportez-vous à la section [Enregistrement sur disque en continu](#) de ce chapitre pour obtenir des informations complémentaires sur la façon de maintenir les fichiers ouverts pendant que 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 14-1 montre comment utiliser le VI de haut niveau Écrire dans un fichier tableur pour envoyer des nombres vers un fichier tableur Microsoft Excel. Lorsque vous exécutez ce VI, LabVIEW vous invite à écrire les données dans un fichier existant ou à créer un nouveau fichier.

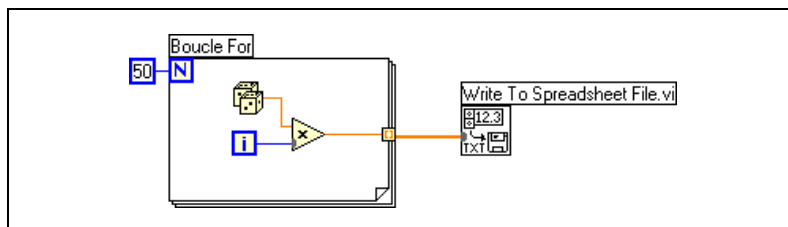


Figure 14-1. Utilisation d'un VI de haut niveau pour écrire dans un fichier tableur

Utilisez les VIs de fichiers binaires pour lire des fichiers binaires et écrire dans ces fichiers. Les données peuvent être des entiers ou des nombres à virgule flottante simple précision.

Utilisation de VIs et de fonctions d'E/S sur fichiers de bas niveau et avancés

Utilisez les VIs et les fonctions de bas niveau et les fonctions d'E/S sur fichiers avancées pour contrôler individuellement chaque opération 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, représenté 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 obtenir de plus amples informations sur les commandes et indicateurs de chemin.

La figure 14-2 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.

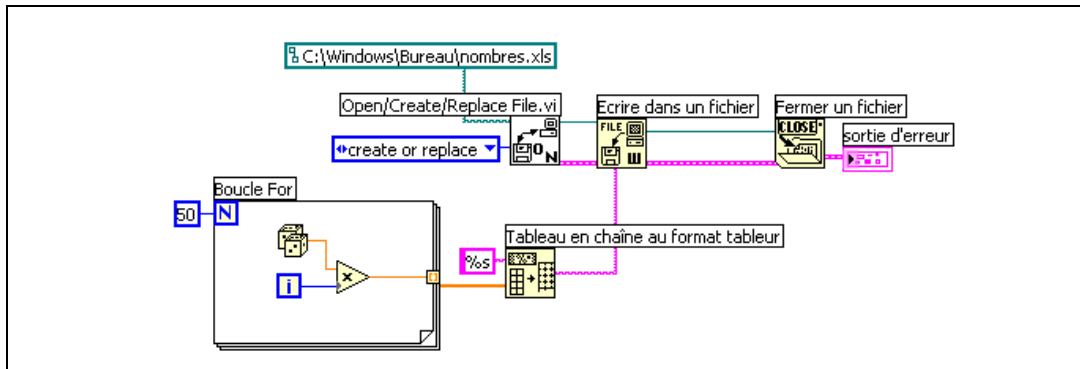


Figure 14-2. Utilisation d'un VI de bas niveau pour écrire dans un fichier tableur

Comparez le VI de la figure 14-2 au VI de la figure 14-1, qui exécute la même tâche. Dans la figure 14-2, 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 14-1 ouvre le fichier, convertit le tableau de nombres en chaîne et ferme le fichier.

Reportez-vous au VI Write Datalog File Example dans `examples\file\datalog.llb` pour consulter 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 VIs et fonctions, tels que les VIs et 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, vous devez convertir vos données en chaîne. Pour écrire des données dans un fichier de type tableur, vous devez formater la chaîne en chaîne de type tableur, c'est-à-dire en chaîne comportant des séparateurs, tels que des tabulateurs. Reportez-vous à la section [Formatage de chaînes](#) du chapitre 10, [Groupement des données au moyen de chaînes, de tableaux et de clusters](#), pour obtenir de plus amples informations 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](#) de ce chapitre pour obtenir de plus amples informations 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 différents styles et différentes polices de caractères, couleurs et tailles que les VIs d'E/S sur fichiers de haut niveau 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* et les fonctions *Tableau* pour formater les données et pour concaténer les chaînes. Écrivez ensuite les données dans un fichier. Reportez-vous au chapitre 10, [Groupement des données au moyen de chaînes, de tableaux et de clusters](#), pour obtenir des informations complémentaires 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. La plupart du temps, vous pouvez utiliser cette fonction au lieu de formater séparément la chaîne avec la fonction *Formater* en chaîne et écrire la chaîne résultante avec le VI *Écrire des caractères* dans un fichier ou la fonction *Écrire* dans un fichier.

Reportez-vous à la section *Formatage de chaînes* du chapitre 10, *Groupement des données au moyen de chaînes, de tableaux et de clusters*, pour obtenir de plus amples informations sur le formatage de chaînes.

Balayage de données à partir de fichiers

Utilisez la fonction *Balayer un fichier* afin de rechercher dans celui-ci des valeurs de chaînes, nombres, chemins et des valeurs booléennes, pour les convertir ensuite en type de données. Vous pouvez souvent utiliser cette fonction plutôt que de lire les données 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*.

Création de fichiers binaires

Pour créer un fichier binaire, vous devez acquérir un ensemble de nombres et les écrire dans un fichier. Utilisez les VIs *Écrire dans un fichier I16* et *Écrire dans un fichier SGL* pour enregistrer des tableaux 1D et 2D de nombres entiers ou de nombres à virgule flottante 16 bits simple précision dans un fichier. 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 différents types de données, tels que des flottants double précision ou des entiers 32 bits non signés, utilisez les fonctions de fichiers de bas niveau ou 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\smp1file.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 dans le menu Exécution ou en utilisant les fonctions de fichier de bas niveau ou 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 des informations complémentaires 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 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](#) de ce chapitre pour obtenir des informations complémentaires sur les fichiers journaux.

Le VI de la figure 14-3 acquiert des signaux multiples, les affiche sous forme de données waveform dans un graphe et les écrit dans un fichier tableur.

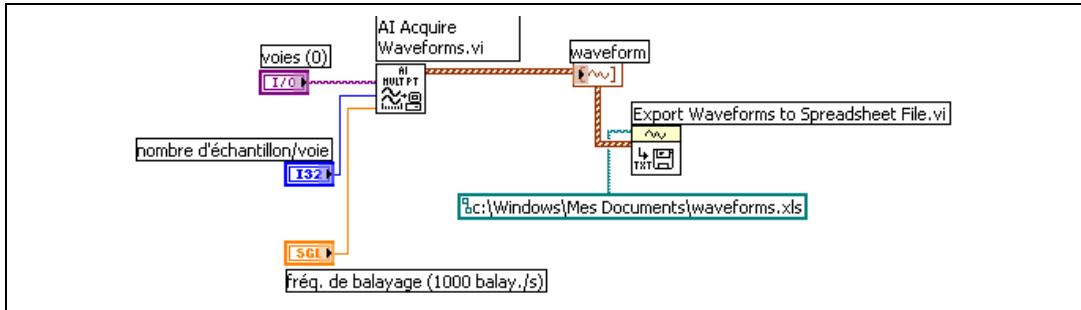


Figure 14-3. Écriture de multiples waveforms dans un fichier tableau

Lecture de données waveform dans des fichiers

Le VI Lire une waveform à partir d'un fichier lit aussi des waveforms multiples dans un fichier. Après avoir lu une seule waveform, vous pouvez ajouter ou modifier des composants de type de données waveform grâce à la fonction Construire une waveform ou vous pouvez extraire des composants de données waveform à l'aide de la fonction Obtenir un attribut de waveform.

Le VI de la figure 14-4 lit des données waveform dans un fichier, édite la composante **dt** et trace les données waveform éditées dans un graphe.

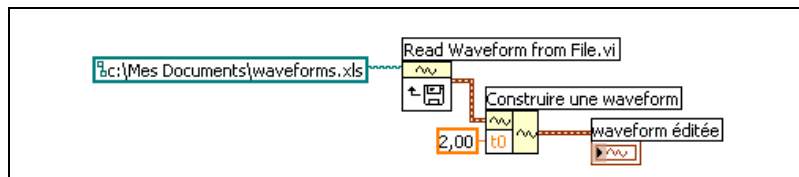


Figure 14-4. Lecture d'une waveform dans un fichier

Le VI Lire une waveform à partir d'un fichier lit aussi des waveforms multiples dans 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 14-5. Reportez-vous à la section [Tableaux](#) du chapitre 10, [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 et structures*, pour obtenir de plus amples informations sur les graphes.

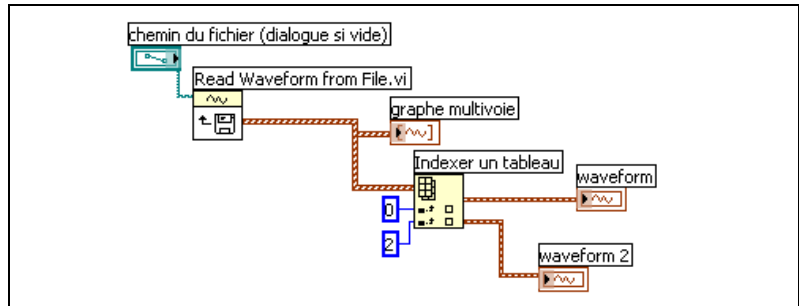


Figure 14-5. Lecture de waveforms multiples dans un fichier

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 artificielle des données* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations sur la dépendance artificielle de données.

Création de fichiers de configuration

Utilisez les 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, comme des chemins, dans un format indépendant de la plate-forme de manière à pouvoir utiliser les fichiers générés par ces VIs sur plusieurs plates-formes. Les VIs de fichiers de configuration n'utilisent pas un format de fichier standard pour les fichiers de configuration. Bien que vous puissiez utiliser les VIs de fichiers de configuration sur n'importe quelle plate-forme pour lire et écrire des

fichiers créés par les VIs, vous ne pouvez pas utiliser les VIs de fichiers de configuration pour créer ou modifier des fichiers de configuration au format Mac OS ou UNIX.

Reportez-vous à `exemples\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, à condition que le contenu du fichier soit au bon format. Reportez-vous à la section *Format de fichier de paramètres de configuration Windows* de ce chapitre pour obtenir des informations complémentaires 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 14-6. Ce VI utilise le VI Lire une clé pour lire la **clé** appelée `Valeur` dans la **section** appelée `Données`. 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.

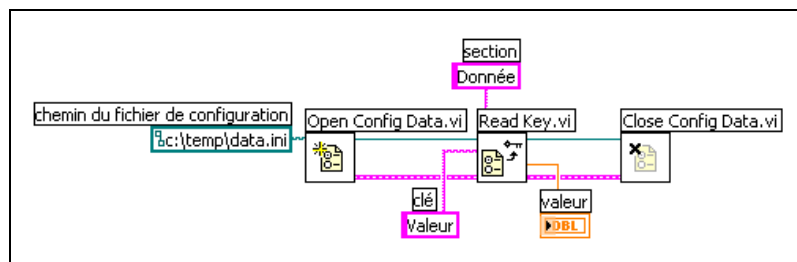


Figure 14-6. Lecture des données à partir d'un fichier `.ini`

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 à VRAI pour les données brutes et à 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
- **(Mac OS)** c:temp:data.dat
- **(UNIX)** /c/temp/data.dat

Les VIs interprètent le chemin relatif temp/data.dat comme suit :

- **(Windows)** temp\data.dat
- **(Mac OS)** :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.

Dès que vous exécutez un VI, le fichier journal de la face-avant enregistre les données qu'elle contient 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 du fichier journal est 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 14-7 pour obtenir un exemple de cette commande numérique.

Journaux de la face-avant automatiques et interactifs

Sélectionnez **Exécution»Enregistrer dans le journal après exécution** pour activer l'enregistrement automatique. La première fois que vous enregistrez dans un fichier journal 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 de graphe déroulant, le fichier journal contient un sous-ensemble du tableau affiché par le graphe déroulant.

Affichage interactif des données de la face-avant enregistrées dans un fichier journal

Après avoir enregistré des données dans un fichier journal, vous pouvez les afficher de manière interactive en sélectionnant **Exécution»Fichier journal»Retrouver...** La barre d'outils de récupération des données s'affiche, comme l'illustre la figure 14-7.

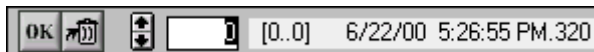


Figure 14-7. Barre d'outils de récupération de données

Le numéro en surbrillance indique quel enregistrement de données vous êtes en train d'afficher. 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 consigné dans le fichier

journal. Affichez 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.

Outre l'affichage de la barre d'outil d'extraction de données, 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, affichez 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 supprimez pas les enregistrements marqués et que vous cliquez sur le bouton **OK**, LabVIEW vous invite à supprimer les enregistrements marqués.

Effacer la liaison vers le fichier journal

Utilisez la liaison vers un fichier journal pour associer un VI au fichier journal qui sera utilisé pour enregistrer ou récupérer des 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**. LabVIEW vous invitera à spécifier le fichier journal la prochaine fois que vous exécuterez le VI en mode d'enregistrement des données automatique ou interactif.

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 avec le bouton droit sur un sous-VI 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 14-8.

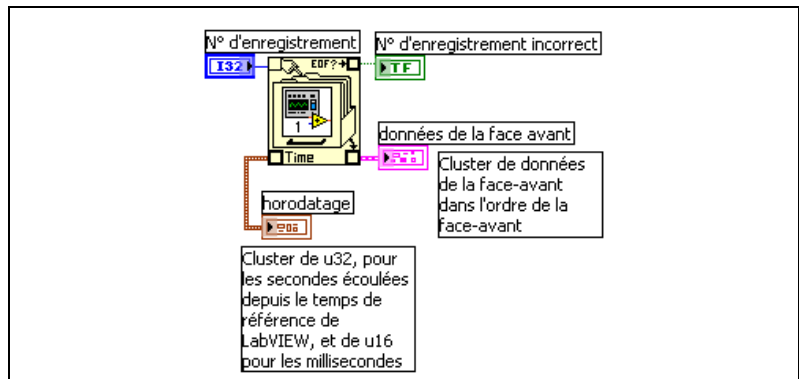


Figure 14-8. Récupération des données enregistrées

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 14-9.

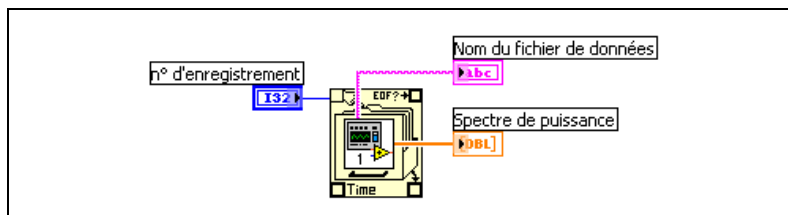


Figure 14-9. Récupération des données enregistrées par l'intermédiaire de terminaux de sous-VIs

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.



Remarque Si vous affichez un sous-VI ou un VI Express VI en tant que nœud extensible, vous ne pouvez pas autoriser l'accès à la base de données pour ce nœud.

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 ayant trait au dernier enregistrement dans le fichier journal en utilisant des 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 numéro n n'appartenant pas à l'intervalle $-n$ à $n - 1$ au terminal **No. d'enregistrement**, la sortie **numéro d'enregistrement incorrect** est VRAI et le sous-VI ne récupère aucune donnée.

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 contient l'horodatage et l'autre 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 système de LabVIEW, c'est-à-dire minuit, le 1er janvier 1904, temps universel.

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 fichiers 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 14-10.

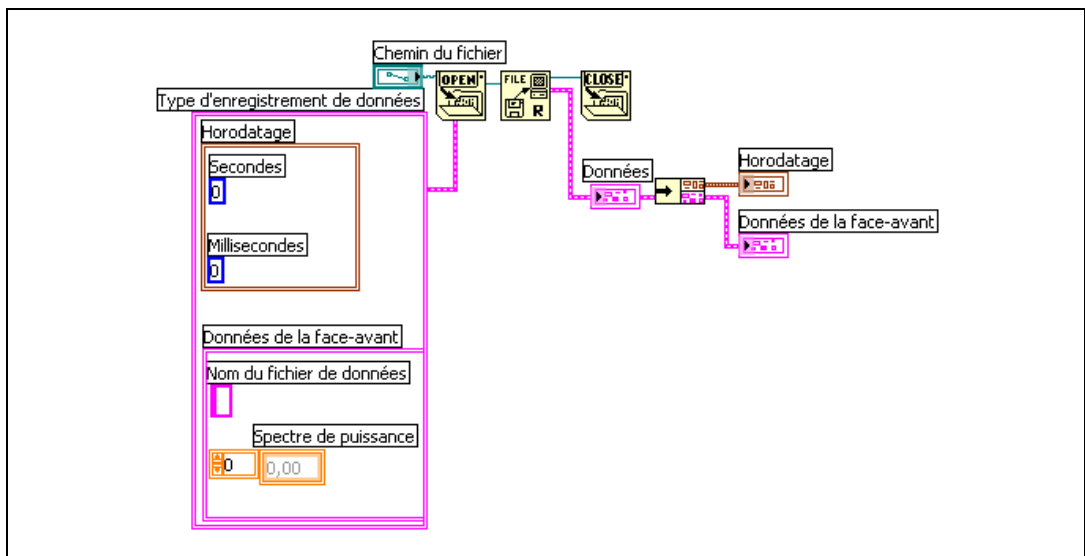


Figure 14-10. Récupération des données enregistrées dans le fichier journal à l'aide de la fonction Ouvrir un fichier

Répertoire de données LabVIEW

Utilisez le répertoire `LabVIEW\data` par défaut pour stocker les fichiers de données générés par LabVIEW, comme les fichiers `.lvnm` ou `.txt`.

LabVIEW installe le répertoire `LabVIEW\data` dans le répertoire par défaut de votre système d'exploitation afin de vous aider à organiser et à localiser les fichiers de données générés par LabVIEW. Par défaut, le VI Express Écrire un fichier de mesures LabVIEW stocke les fichiers `.lvnm` qu'il génère dans ce répertoire et le VI Express Lire un fichier de mesures LabVIEW lit ce répertoire. La constante Répertoire de données par défaut, représentée à gauche, et la propriété Répertoire de données par défaut renvoient elles aussi le répertoire `LabVIEW\data` par défaut.



Sélectionnez **Outils»Options** et sélectionnez ensuite **Chemins** dans le menu déroulant du haut de la fenêtre pour spécifier un autre répertoire de données par défaut. Le répertoire de données par défaut est différent du répertoire par défaut, qui est le répertoire que vous spécifiez pour les nouveaux VIs, les commandes personnalisées, les modèles de VIs et autres documents LabVIEW que vous créez.

Fichier de données de mesures LabVIEW

Le fichiers de données de mesures LabVIEW (`.lvnm`) contient les données générées par le VI Express Écrire un fichier de mesures LabVIEW. Le fichier de données LabVIEW est un fichier texte délimité par des tabulations que vous pouvez ouvrir avec une application de type tableur ou un éditeur de texte. En plus des données générées par un VI Express, le fichier `.lvnm` contient des informations relatives aux données, telles que la date et l'heure auxquelles les données ont été générées.

Reportez-vous au manuel *Initiation à LabVIEW* pour obtenir des informations complémentaires sur la manière d'enregistrer et de récupérer des données à l'aide des VIs Express.

Vous pouvez utiliser un fichier de données de mesures LabVIEW avec NI DIAdem.

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.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur la documentation et l'impression des VIs, ainsi que sur la génération de rapports.

Documentation des VIs

Vous pouvez utiliser LabVIEW pour faire le suivi du développement, documenter un VI achevé et créer des instructions destinées aux utilisateurs des VIs. Vous pouvez afficher la documentation sous LabVIEW, l'imprimer et la sauvegarder sous forme de fichiers HTML, RTF ou texte.

Pour créer une documentation utile sur des VIs, ajoutez des commentaires à l'historique des révisions des VIs et créez des descriptions de VIs, ainsi que des descriptions d'objets.

Activation de l'historique des révisions des VIs

Utilisez la fenêtre **Historique** pour afficher l'historique du développement d'un VI, notamment les numéros de révision. Lorsque vous apportez des modifications au VI, notez celles-ci et faites-en un suivi dans la fenêtre **Historique**. Sélectionnez **Outils»Historique des révisions du VI** pour ouvrir la fenêtre **Historique**. Vous pouvez aussi imprimer l'historique des révisions et l'enregistrer dans un fichier texte, HTML ou RTF.

Reportez-vous à la section [Impression de la documentation](#) de ce chapitre pour obtenir des informations complémentaires sur l'impression de l'historique des révisions du VI ou sur son enregistrement dans un fichier.

Numéros de révision

Le numéro de révision permet de suivre 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. Pour afficher le numéro de révision actuel dans la barre de titre du VI et la barre de titre de la fenêtre **Historique**, sélectionnez **Outils»Options**, sélectionnez **Historique des révisions** dans le menu déroulant du haut et cochez la case **Afficher le numéro de révision dans la barre de titre**.

Le numéro qu'affiche LabVIEW dans la fenêtre **Historique** est le prochain numéro de révision, c'est-à-dire le 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.

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 afficher les descriptions dans LabVIEW, les imprimer ou les enregistrer dans des fichiers texte, HTML ou RTF.

Créez, éditez et affichez les descriptions des VIs en sélectionnant **Fichier» Propriétés du VI** et en sélectionnant **Documentation** dans le menu déroulant **Catégorie**. Créez, éditez et affichez 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 quand un VI est en cours d'exécution. 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.

Impression de la documentation

Sélectionnez **Fichier»Imprimer...** pour imprimer la documentation d'un VI ou l'enregistrer dans un fichier texte, 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 :

- Icône et connecteur
- Face-avant et diagramme
- Commandes, indicateurs et terminaux de types de données
- VI et descriptions d'objets
- Hiérarchie du VI
- Liste des sous-VIs
- Historique des révisions



Remarque La documentation que vous créez pour certains types de VIs ne peut pas inclure tous les éléments précédents. Par exemple, comme un VI polymorphe ne possède pas de face-avant ni de diagramme, vous ne pouvez pas inclure ces éléments dans la documentation que vous créez pour un VI polymorphe.

Enregistrement de la documentation sous forme de fichiers texte, HTML ou RTF

Vous pouvez enregistrer la documentation du VI sous forme de fichiers texte, HTML ou 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 à la section *Impression de VIs par programmation* de ce chapitre afin d'obtenir de plus amples informations sur l'impression et l'enregistrement de la documentation par programmation dans des fichiers texte, HTML et RTF.

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, PNG ou GIF.

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 des couleurs réparties inégalement. Les graphiques JPEG sont toujours des graphiques 24 bits. Si vous sélectionnez une profondeur de couleurs inférieure, comme du 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 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. 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 offre certains avantages par rapport aux formats JPEG et GIF, il n'est pas aussi bien pris en charge par les 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. Pour obtenir des fichiers GIF compressés de meilleure qualité, sélectionnez le format PNG lorsque vous enregistrez la documentation et utilisez un convertisseur de format d'image pour convertir les fichiers PNG enregistrés par LabVIEW en fichiers GIF. Une sélection initiale du format PNG produit des graphiques de plus haute qualité car le format PNG est la reproduction exacte de l'image d'origine. Modifiez le fichier HTML généré par LabVIEW pour faire référence aux fichiers GIF avec l'extension `.gif`.

Conventions d'appellation des fichiers image

Lorsque vous générez de la documentation HTML ou RTF qui comporte des images externes, LabVIEW enregistre les terminaux de type de données des commandes et indicateurs dans des fichiers image dont les noms sont compatibles. Si un VI contient plusieurs terminaux du même type, LabVIEW crée un seul fichier image pour ce type-là. Par exemple, si un VI a trois entiers signés 32 bits en entrée, LabVIEW crée un seul fichier `ci32.x`, où `x` représente l'extension correspondant au format graphique.

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, (**Mac OS**) 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 :

- 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 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 17, [Contrôle des VIs par programmation](#), pour obtenir de plus amples informations sur l'utilisation de cette méthode pour imprimer les 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 la zone de travail de la fenêtre active, y compris tous les objets qui figurent dans la portion non visible de la fenêtre. 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 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 de VIs par programmation](#) de ce chapitre pour obtenir des informations complémentaires sur l'impression par programmation.

Impression de VIs par programmation

Utilisez les méthodes suivantes pour imprimer des VIs par programmation plutôt que de manière interactive au moyen des boîtes de dialogue qui s'affichent lorsque vous sélectionnez **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 les VIs de génération de rapports pour imprimer des rapports ou enregistrer des rapports HTML contenant la documentation du VI ou les données renvoyées par le VI.
- Utilisez le VI Serveur pour imprimer par programmation la fenêtre d'un VI ou la documentation d'un VI, ou pour l'enregistrer dans un fichier texte, HTML ou RTF à n'importe quel moment. Reportez-vous au chapitre 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur l'utilisation de cette méthode pour imprimer les VIs.



Remarque Si vous imprimez la documentation d'un VI à partir d'une application construite, vous ne pouvez imprimer que les faces-avant.

Impression de la face-avant d'un VI après exécution du VI

Sélectionnez **Exécution»Imprimer à la fin de l'exécution** pour imprimer la face-avant d'un VI une fois l'exécution du VI terminée. 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**.

Sélectionner ces options revient à sélectionner **Fichier»Imprimer la fenêtre** lorsque la face-avant est la fenêtre active.

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 imprimer des données à partir d'un VI de niveau principal

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. Au lieu de sélectionner **Exécution** » **Imprimer à la fin de l'exécution** dans le VI de niveau principal, sélectionnez cette option dans le sous-VI. Lorsque vous voulez imprimer, appelez le sous-VI et transmettez-lui les données à imprimer.

Génération et impression de rapports

Utilisez les VIs de génération de rapports pour imprimer des rapports ou enregistrer des rapports HTML contenant la documentation du VI ou les données renvoyées par le VI. Utilisez le VI Impression simple de face-avant de VI pour générer un rapport de base contenant de la documentation sur le VI. Utilisez le VI Rapport texte facile pour générer un rapport de base contenant les données renvoyées par le VI. Utilisez les autres VIs de génération de rapport pour générer des rapports plus complexes.



Remarque Vous ne pouvez générer des rapports que 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 :

- Ajouter du texte, des images, des tables ou de la documentation sur un VI à un rapport.
- Définir la police, la taille, le style et la couleur du texte.
- Définir l'orientation du rapport (portrait ou paysage).
- Définir les en-têtes et les pieds de page du rapport.
- Définir les marges et les tabulations.

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, Mac OS X et UNIX)** Utilisez le VI System Exec.
- **(Mac OS)** Utilisez le VI AESend Print Document.
- **(Windows)** Utilisez ActiveX pour faire imprimer des données par une autre application. Reportez-vous au chapitre 19, [Connectivité Windows](#) pour obtenir de plus amples 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 17, *Contrôle des VIs par programmation*, pour obtenir des informations complémentaires sur l'utilisation du VI Serveur pour configurer la manière dont un VI se comporte.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations 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 aussi utiliser cette page pour modifier l'icône ou la taille de la grille d'alignement pour le VI.
- **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 15, *Documenter et imprimer des VIs*, pour obtenir des informations complémentaires 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 de plus amples informations sur la création de VIs multithread.
- **Options d'édition** : utilisez cette page pour définir la taille de la grille d'alignement du VI actuel et pour changer le style de la commande ou de l'indicateur créé par LabVIEW lorsque vous cliquez avec le bouton droit sur un terminal et que vous sélectionnez **Créer»Commande** ou **Créer»Indicateur** dans le menu local. Reportez-vous à la section *Alignement et répartition d'objets* du chapitre 4, *Construction de la face-avant*, pour obtenir de plus amples informations sur l'alignement sur grille.

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 **Afficher 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 (.rtm) 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 de plus amples 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 menus pour personnaliser les 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 figure 16-1, la fonction Élément de menu sélectionné lit l'élément de menu sélectionné par l'utilisateur et transmet cet élément de menu à la structure Condition où il va s'exécuter.

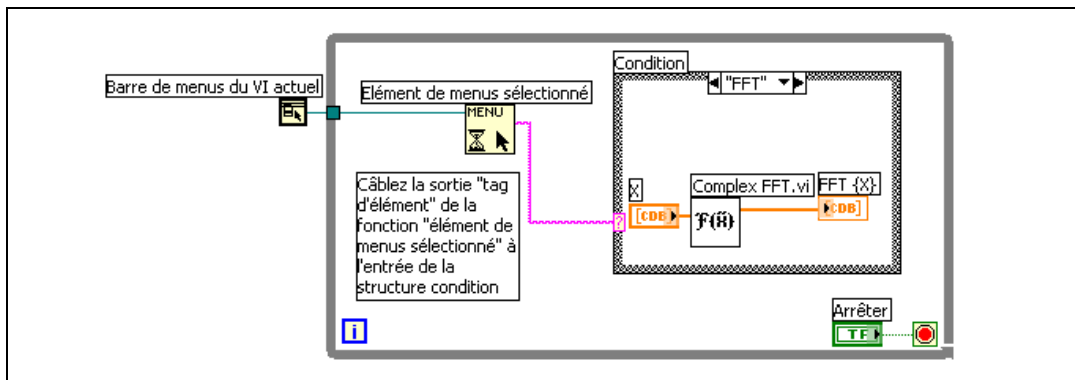


Figure 16-1. Diagramme utilisant la gestion des menus

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.

Vous pouvez aussi utiliser la structure Événement pour gérer des événements de menus. Reportez-vous au chapitre 9, *Programmation événementielle*, pour obtenir de plus amples informations sur la structure Événement.

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 obtenir 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 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 de plus amples informations sur les refnums.

Normalement, 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 nœuds de contrôle d'applications suivants pour construire des applications de type VI Serveur :

- **Ouvrir une référence d'application** : ouvre une référence à l'application locale ou l'application distante à laquelle vous avez accès par l'intermédiaire du serveur ou pour accéder à une instance distante de LabVIEW.

- **Ouvrir une référence de VI** : Ouvre une référence à un VI sur l'ordinateur local ou distant, ou charge un VI de manière dynamique à partir d'un disque.
- **Nœud de propriété** : obtient et définit les propriétés du VI, de l'objet ou de l'application. Reportez-vous à la section *Nœuds de propriété* de ce chapitre pour obtenir de plus amples informations sur les propriétés.
- **Nœud de méthode** : appelle les méthodes sur un VI, un objet ou une application. Reportez-vous à la section *Nœuds de méthode* de ce chapitre pour obtenir de plus amples informations sur les méthodes.
- **Nœud d'appel par référence** : appelle un VI chargé dynamiquement.
- **Fermer la référence** : ferme la référence ouverte à un VI, un objet ou une application à laquelle vous avez accédé au moyen du VI Serveur.

Références d'application et de 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 une 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 créez un refnum pour un VI, LabVIEW charge le VI en mémoire. Ce 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 la documentation du 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 de 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 à l'aide d'un seul nœud ; toutefois, certaines propriétés ne peuvent pas être modifiées par écriture. 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.

Si le nœud de propriété ouvre et renvoie une référence à une application ou à un VI, utilisez la fonction Fermer la référence pour fermer la référence à l'application ou au VI. LabVIEW ferme la référence à une commande chaque fois que cette référence n'est plus nécessaire. Vous n'avez pas besoin de fermer vous-même les références à la commande.

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 de la face-avant. 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 au terminal de l'objet de la face-avant ou à la référence de commande. Reportez-vous à la section *Contrôle des objets de la face-avant* de ce chapitre pour obtenir de plus amples informations sur les références 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 renvoie une valeur, le terminal de la méthode affiche la valeur renvoyée. 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 affiché en haut et les paramètres optionnels grisés affichés en bas.

Manipulation des propriétés et des méthodes de classe 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 17-1 montre comment afficher tous les VIs en mémoire sur un ordinateur local dans un tableau de chaînes sur la face-avant.

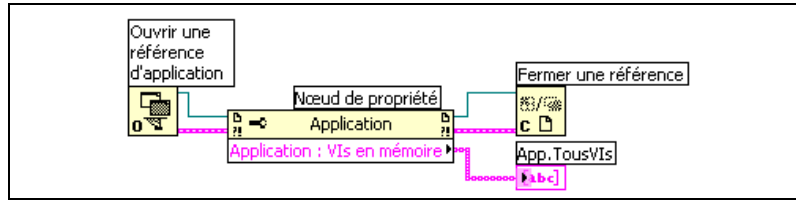


Figure 17-1. Affichage de tous les VIs en mémoire sur un ordinateur local

Si vous ne câblez pas de refnum à l'entrée **référence**, le nœud de propriété ou le nœud de méthode utilise une référence à l'instance en cours de LabVIEW. Si vous voulez manipuler les propriétés ou les méthodes d'une autre instance de LabVIEW, vous devez câbler un refnum d'application à l'entrée **référence**.

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 par la figure 17-2, 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 17-1 ne s'applique qu'aux instances locales de LabVIEW.

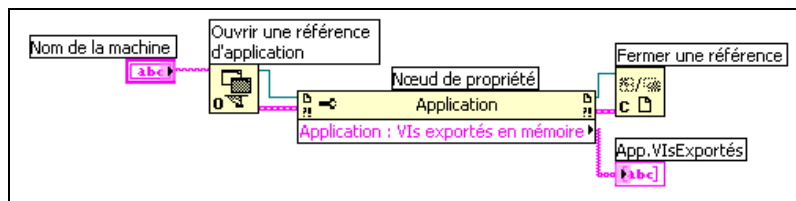


Figure 17-2. Affichage de tous les VIs en mémoire sur un ordinateur à distance

Manipulation des propriétés et des méthodes de classe VI

Vous pouvez obtenir ou définir les propriétés d'un VI, effectuer des méthodes sur un VI, ou bien les deux. Dans la figure 17-3, 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 s'ouvre et affiche les valeurs par défaut au moyen du nœud de propriété.

Si vous ne câblez pas de refnum à l'entrée **référence**, le nœud de propriété ou le nœud de méthode utilise une référence au VI qui contient le nœud de propriété ou le nœud de méthode. Si vous voulez manipuler les propriétés ou les méthodes d'un autre VI, vous devez câbler un refnum de VI à l'entrée **référence**.

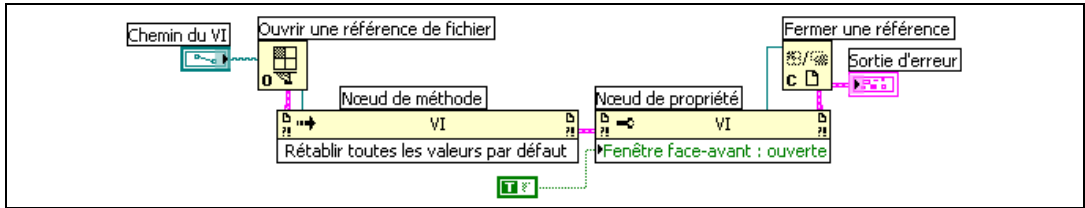


Figure 17-3. Utilisation des nœuds de méthode et de propriété de classe VI

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 VI.

Manipulation des propriétés et des méthodes de classe Application et VI

Pour certains VIs, vous devez accéder aux propriétés et aux méthodes de classe Application et VI en même temps. Vous devez ouvrir et fermer les refnums de classe Application et VI séparément, comme l'illustre la figure 17-4.

La figure 17-4 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 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'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.

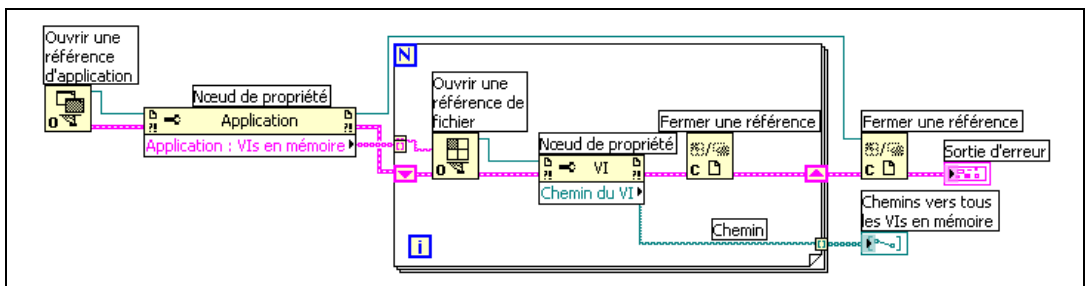


Figure 17-4. Utilisation des propriétés et méthodes d'application et de classe VI

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 exige 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 17-5 montre comment se servir du nœud d'appel par référence pour appeler le VI de réponse en fréquence de façon dynamique. Le nœud d'appel par référence nécessite l'utilisation des fonctions Ouvrir une référence de VI et Fermer la référence, similaires aux fonctions que vous utilisez pour le nœud de propriété et le nœud de méthode.

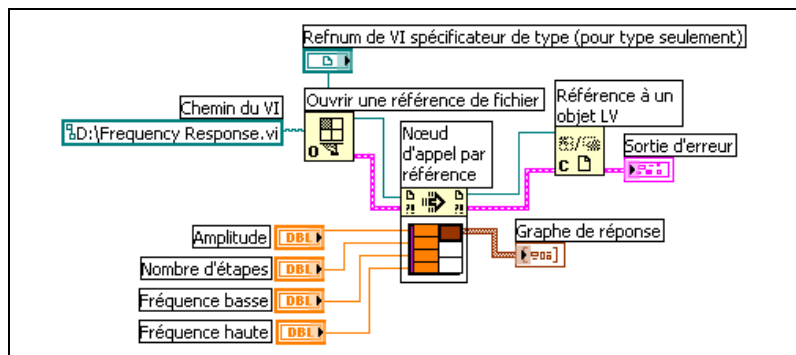


Figure 17-5. Utilisation du nœud d'appel par référence

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 renvoie 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

Les références de commandes correspondent aux références d'objets dans les interfaces utilisateur des langages de programmation textuels. Les références de commandes ne correspondent pas aux pointeurs dans les langages de programmation textuels.

Utilisez la commande Refnum de commande de refnum située sur les palettes **Refnum** et **Refnum classique** pour transmettre des références d'objets de la face-avant à d'autres VIs. Vous pouvez aussi cliquer avec le bouton droit sur un objet de la face-avant et sélectionner **Créer»Référence** dans le menu local pour créer une référence de commande. Une fois que vous avez transmis une référence de commande à un sous-VI, utilisez les nœuds de propriété et les nœuds de méthode pour lire et écrire les propriétés et appeler les méthodes d'un objet référencé de la face-avant.

Reportez-vous à la section *Structures Condition et Séquence* du chapitre 8, *Boucles et structures*, 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 exactement. Par exemple, si le type d'un refnum de commande de type strict est une glissière d'entiers 32 bits, vous ne pouvez câbler qu'une glissière d'entiers 32 bits au terminal du refnum de commande. Vous ne pouvez pas câbler une glissière d'entiers 8 bits, une glissière scalaire double précision, ni un cluster de glissières d'entiers 32 bits au terminal du refnum de commande.

Par défaut, les références de commande que vous créez à partir d'une commande sont de type strict. Une étoile rouge dans le coin inférieur gauche de la référence de commande sur la face-avant indique que la référence 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 de la référence de commande pour indiquer que la référence 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 références de commande partiellement spécifiées.

Les références de commande partiellement spécifiées sont plus souples au niveau du type de données qu'elles acceptent. Ainsi, si le type d'une référence de commande partiellement spécifiée 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 de la référence de commande. Si le type d'une référence de commande partiellement spécifiée est une commande, vous pouvez câbler une référence de commande de tout type de commande au terminal de la référence de commande.



Remarque Lorsque vous câblez un nœud de propriété à un terminal de référence 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 la référence de graphe déroulant est de type strict. Reportez-vous à la section [Manipulation de données variant](#) du chapitre 5, [Construction du diagramme](#), pour obtenir de plus amples informations 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 fonctionnalités 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 la face-avant et de la documentation de VIs sur le Web.
- Envoyer des e-mails à partir de VIs.
- Construire des VIs qui communiquent avec d'autres applications et d'autres VIs au moyen de protocoles de bas niveau, tels que TCP, UDP, Apple Events et PPC Toolbox.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations complémentaires sur la mise en réseau dans LabVIEW.

Choisir une solution parmi 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 pouvant être lues par d'autres VIs et d'autres applications, utilisez les VIs et les fonctions d'E/S sur fichiers. Reportez-vous au chapitre 14, *E/S sur fichiers*, pour obtenir de plus amples informations sur 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 17, *Contrôle des VIs par programmation*, pour obtenir de plus amples informations sur le contrôle des VIs et des autres applications LabVIEW sur des ordinateurs locaux ou distants.

(Windows) Pour accéder aux fonctions de nombreuses applications de Microsoft, telles que l'intégration d'un graphe waveform dans un fichier tableur Excel, utilisez les VIs et les fonctions ActiveX. Reportez-vous au chapitre 19, *Connectivité Windows*, 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 17, *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. HTTP et TCP/IP font partie des protocoles que vous pouvez utiliser. Vous choisissez le 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* de ce chapitre si vous souhaitez obtenir des informations complémentaires sur les protocoles de communication supportés par LabVIEW.

(Windows) Reportez-vous au chapitre 19, *Connectivité Windows*, si vous souhaitez obtenir des informations complémentaires 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 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 ouvrir 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 spécifiez des URL dans un navigateur Web.

Par exemple, pour partager les données d'un indicateur 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 *Integrating the Internet into Your Measurement System* (Intégration d'Internet dans votre système de mesure) pour obtenir 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 `file`, 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 :

- **DataSocket Transport Protocol** (`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, ce tag s'ajoute à l'URL. La connexion DataSocket l'utilise pour accéder à un élément de données particulier sur un serveur DataSocket. Un serveur DataSocket doit être lancé pour pouvoir utiliser ce protocole.

- **(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, un serveur OPC doit être lancé.
- **(Windows) logos** : une technologie interne à National Instruments pour transmettre des données entre le réseau et votre ordinateur local.
- **File Transfer Protocol (ftp)** : vous pouvez utiliser ce protocole pour spécifier un fichier à partir duquel vous voulez lire des données.



Remarque Pour lire un fichier texte à partir d'un site FTP à l'aide du protocole DataSocket, ajoutez [text] à la fin de l'URL DataSocket.

- **file** : vous pouvez utiliser ce protocole pour fournir un lien vers un fichier, local ou sur un réseau, qui contient des données.

La table 18-1 fournit des exemples d'URL pour chaque protocole.

Tableau 18-1. Exemples d'URL DataSocket

URL	Exemple
dstp	dstp://nomserveur.com/numerique, où numerique est le tag nommé pour les données
opc	opc:\National Instruments.OPCTest\element1 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://nom_ordinateur/processus/nom_element_donnée
ftp	ftp://ftp.ni.com/datasocket/ping.wav
file	file:ping.wav file:c:\donnee\ping.wav file:\\machine\donnee\ping.wav

Utilisez les URL `dstp`, `opc` et `logos` pour partager des données en direct, car ces protocoles peuvent mettre à jour les commandes et les indicateurs distants et locaux. Utilisez les URL `ftp` et `file` pour lire les données des fichiers, car ces protocoles ne peuvent pas mettre à jour les commandes et les indicateurs distants et locaux.

Reportez-vous à `exemples\comm\datasocket.llb` pour consulter des exemples d'utilisation de 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 fournir une chaîne à un indicateur chaîne.
- **Texte tabulé** : utilisez du texte tabulé, comme celui d'un tableur, 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'un contrôle ActiveX National Instruments Measurement Studio.

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 affichent 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 une image de la 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 scénarios 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 dans une commande câblée à un sous-VI ou à une fonction, ou bien il peut afficher les données dans un indicateur.
- Publiez une valeur qui apparaît dans un indicateur de la face-avant, afin qu'un autre utilisateur puisse souscrire aux données et les afficher 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 d'un sous-VI ou d'une fonction. Par exemple, un VI qui calcule la température moyenne et l'affiche dans un thermomètre sur la face-avant peut publier la donnée température.
- Souscrivez à une valeur qui apparaît dans une commande ou un indicateur sur la face-avant d'un autre VI pour afficher les données dans un indicateur de la face-avant de votre VI. Si vous souscrivez aux données par l'intermédiaire d'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 nécessairement 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, être convertibles. Par exemple, vous pouvez utiliser un indicateur numérique de votre VI pour

afficher 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, des serveurs FTP ou des serveurs Web, utilisez la fonction DataSocket Lire, les fonctions et les VIs d'E/S sur fichiers ou les fonctions et les VIs de contrôle d'applications.

Lecture et écriture de données en direct depuis le diagramme

Dans le diagramme, vous pouvez lire ou écrire des données par programmation à l'aide des fonctions DataSocket.

Utilisez la fonction DataSocket Écrire pour écrire des données en direct en utilisant une connexion DataSocket par programmation. La figure 18-1 explique comment écrire une valeur numérique.

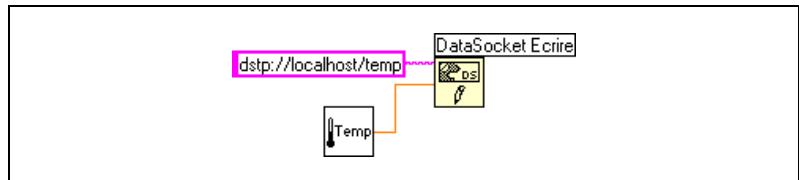


Figure 18-1. Publication de données à l'aide de DataSocket Écrire

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 par programmation à partir d'une connexion DataSocket. La figure 18-2 montre comment lire des données et les convertir en un nombre à virgule flottante double précision.

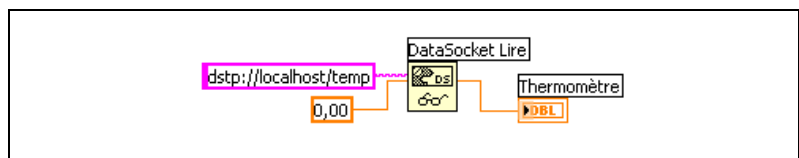


Figure 18-2. Lecture d'une valeur unique à l'aide de DataSocket Lire

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 variant que vous devez manipuler à l'aide de la fonction Variant en donnée.

Ouverture et fermeture de connexions DataSocket par programmation

Utilisez les fonctions DataSocket Ouvrir et DataSocket Fermer pour contrôler à quel moment une connexion DataSocket s'ouvre et se ferme. Lorsque vous ouvrez une connexion DataSocket à l'aide de la fonction DataSocket Ouvrir, la connexion reste ouverte jusqu'à ce que l'une des conditions suivantes soit remplie : vous fermez explicitement la connexion à l'aide de la fonction DataSocket Fermer, vous fermez le VI ou l'exécution du VI s'arrête. L'entrée **URL** de la fonction DataSocket Ouvrir accepte uniquement une URL DataSocket. La fonction DataSocket Ouvrir renvoie un refnum de connexion DataSocket que vous pouvez utiliser comme entrée **URL** des fonctions DataSocket Lire et DataSocket Écrire.

Buffériser des données DataSocket

Si vous utilisez le protocole DataSocket Transport Protocol (`dstp`), le serveur DataSocket publie par défaut uniquement la valeur la plus récente à tous les clients. Lorsqu'un client publie des valeurs sur le serveur plus rapidement qu'un autre client ne les lit, les nouvelles valeurs écrasent les anciennes valeurs non traitées avant que celles-ci ne soient lues par les clients. Cette perte de données non traitées peut se produire au niveau du serveur ou au niveau du client. Il se peut que cette perte de données ne vous pose pas de problème si vous souscrivez aux données DataSocket et que vous voulez recevoir uniquement la valeur la plus récente publiée sur le serveur. Toutefois, si vous voulez recevoir chaque valeur publiée sur le serveur, vous devez buffériser les données sur le client.



Remarque La bufférisation côté client s'applique aussi à d'autres protocoles, comme `opc`, `logos` et `file`. Pour utiliser la bufférisation `dstp`, vous devez également utiliser le DataSocket Server Manager pour configurer la bufférisation côté serveur. Reportez-vous à la rubrique *DataSocket Help* pour obtenir des informations complémentaires sur la bufférisation côté serveur.

La bufférisation `dstp` ne garantit pas la distribution des données. Si les données d'un buffer au niveau du serveur ou du client dépassent la taille du buffer, celui-ci se débarrasse des anciennes valeurs et les remplace par des nouvelles. Pour détecter des valeurs qui ont été éliminées dans un flux de

données, câblez les données publiées à la fonction Définir un attribut de variant pour identifier chaque valeur de manière unique au niveau de la publication et recherchez les numéros d'identification manquants dans les séquences reçues au niveau de la souscription.

Définissez l'entrée **mode** de la fonction DataSocket Ouvrir à **BufferedRead** (lecture bufférisée) ou **BufferedReadWrite** (lecture/écriture bufférisée) et utilisez un nœud de propriété pour définir les propriétés DataSocket pour la taille d'un buffer FIFO. Ceci garantit que LabVIEW enregistre les valeurs que le client reçoit dans un buffer au lieu de les écraser chaque fois que la valeur change.



Remarque Si vous utilisez les propriétés DataSocket pour définir la taille d'un buffer FIFO, vous devez définir l'entrée **mode** de la fonction DataSocket Ouvrir à **BufferedRead** ou **BufferedReadWrite**. Sinon, l'élément au niveau du serveur ne sera pas bufférisé pour la connexion.

La figure 18-3 utilise la bufférisation DataSocket.

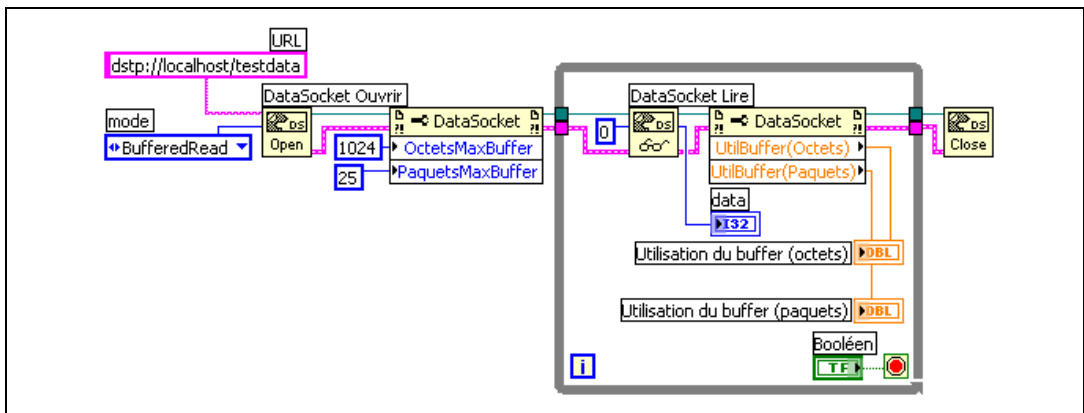


Figure 18-3. Technique de bufférisation DataSocket



Remarque La technique de bufférisation s'applique uniquement quand vous utilisez la fonction DataSocket Lire pour souscrire aux données que le serveur publie. Cette technique ne s'applique pas quand vous utilisez les connexions DataSocket de la face-avant pour souscrire aux données.

Rapport de diagnostic

Utilisez la propriété Utilisation du buffer (octets) ou la propriété Utilisation du buffer (paquets) pour demander des informations de diagnostic sur les buffers que vous spécifiez. Utilisez ces propriétés pour vérifier le

pourcentage de buffer utilisé coté client afin de déterminer si la taille du buffer actuel est adéquate. Si les valeurs de l'une de ces propriétés est proche de la valeur maximale du buffer, augmentez la taille du buffer pour vous assurer que vous recevez bien toutes les valeurs publiées par le serveur.

Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations complémentaires sur la manière de spécifier une taille de buffer pour un client DataSocket.

DataSocket et données variant

Dans certains cas, par exemple lorsque vous souscrivez aux données à partir d'une autre application, le VI ou l'autre application qui lit les données par programmation ne peut pas reconverter les données à leur type initial. Il se peut que vous vouliez aussi 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 pour convertir en données variant, par programmation, les données que vous écrivez dans une connexion DataSocket. La figure 18-4 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.

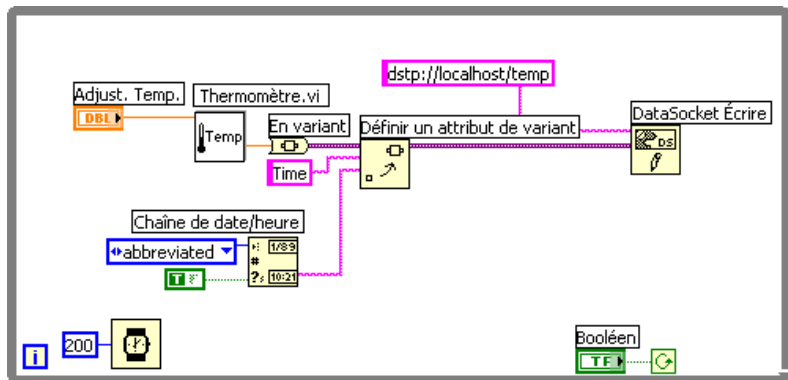


Figure 18-4. Conversion de données de température en direct en données variant

Quand un autre VI lit les données en direct, le VI doit convertir les données variant en un type de données qu'il peut manipuler. La figure 18-5 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.

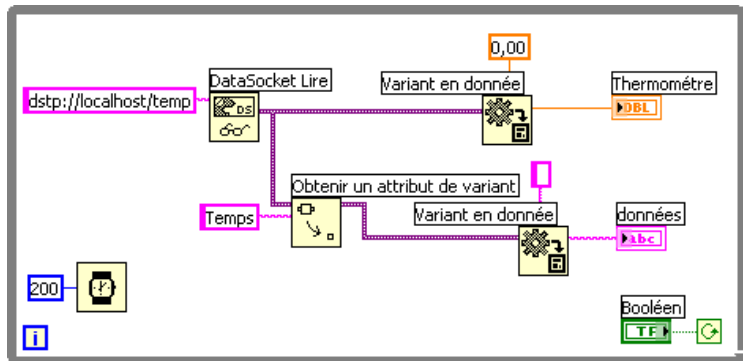


Figure 18-5. Conversion de données variant en direct en données de température

Publication de VIs sur le Web

Utilisez le serveur Web de LabVIEW pour créer des documents HTML, publier des images de faces-avant sur le Web et intégrer des VIs dans une page 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. Consultez le site Web de National Instruments à l'adresse ni.com si vous souhaitez obtenir des informations complémentaires sur ce toolset.

Options du serveur Web

Sélectionnez **Outils»Options** et sélectionnez l'un des éléments **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 des VIs.
- 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 *Création de documents HTML* de ce chapitre. 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 afficher 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.
- Afficher un aperçu du document.
- Enregistrer le document sur disque.
- Activer le serveur Web afin de pouvoir publier des documents HTML et images de face-avant sur le Web.

Publication d'images de face-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://adresse.serveur.web/.snap?Nom.vi`, où *Nom.vi* est le nom du VI que vous voulez afficher.

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://adresse.serveur.web/.monitor?Nom.vi`, où *Nom.vi* est le nom du VI que vous voulez visualiser.

Formats des images de face-avant

Le serveur Web peut générer des images des faces-avant au format d'image JPEG et PNG.

Le format JPEG compresse bien 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.

Affichage 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 vous 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 obtenir de plus amples informations sur l'utilisation du serveur DataSocket.

Configuration du serveur pour les clients

Coté 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 pour fixer une limite au temps de contrôle autorisé à un client distant lorsque d'autres clients sont en attente du contrôle du VI.

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 serveur, reprendre le contrôle de n'importe quel VI à tout moment. Quand l'utilisateur qui contrôle le VI 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 la 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 sous contrôle reflétera ces modifications.

Licence pour face-avant distante

Vous devez configurer une licence pour supporter le nombre de clients potentiels qui peuvent se connecter à votre serveur.



Remarque (Windows 9x) Le système de développement complet LabVIEW (Full Development System) et l'Application Builder comprennent une licence de face-avant distante qui permet à un client d'afficher et de contrôler une face-avant à distance. Le système de développement professionnel LabVIEW comprend une licence de face-avant qui permet à cinq clients d'afficher et de contrôler une face-avant à distance. Vous ne pouvez pas mettre à jour la licence de face-avant distante afin de prendre en charge un plus grand nombre de clients sous Windows 9x.

(Windows 2000/NT/XP, Mac OS et UNIX) Le système de développement complet LabVIEW et l'Application Builder comprennent une licence de face-avant distante qui permet à un client d'afficher et de contrôler une face-avant à distance. Le système de développement professionnel LabVIEW comprend une licence de face-avant qui permet à cinq clients d'afficher et de contrôler une face-avant à distance. Vous pouvez mettre à jour la licence de face-avant distante afin de prendre en charge un plus grand nombre de clients.

Affichage et contrôle de faces-avant dans LabVIEW ou à partir d'un navigateur Web

Un client peut afficher et contrôler une face-avant à distance à partir de LabVIEW ou d'un navigateur Web.



Remarque Avant de pouvoir afficher et contrôler une face-avant à distance, vous devez activer le serveur Web sur l'ordinateur serveur sur lequel se trouve le VI que vous souhaitez afficher.

Affichage et contrôle des faces-avant dans LabVIEW

Pour afficher une face-avant distante en utilisant LabVIEW comme client, ouvrez un nouveau VI et sélectionnez **Exécution»Connexion à une 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 afficher. 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** 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 ne possède le contrôle à ce moment-là, un message apparaît vous indiquant que vous avez le contrôle de la face-avant. En revanche, si un autre client contrôle le VI à ce moment-là, le serveur gardera votre demande dans une file d'attente jusqu'à ce que l'autre client abandonne le contrôle ou que la limite de temps de contrôle soit dépassée. Seul l'utilisateur de l'ordinateur serveur peut surveiller la liste des clients dans la file d'attente en sélectionnant **Outils»Gestionnaire de connexions de face-avant distante**.

Si vous souhaitez sauvegarder les données générées par un VI s'exécutant sur un ordinateur distant, utilisez DataSocket ou TCP à la place des faces-avant distantes. Reportez-vous à la section [Utilisation de la technologie DataSocket](#) de ce chapitre pour obtenir de plus amples informations sur l'utilisation de DataSocket. Reportez-vous à la section [TCP et UDP](#) de ce chapitre pour obtenir de plus amples informations sur l'utilisation de TCP.

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 la face-avant distante** affiche un message d'erreur.

Affichage et contrôle de faces-avant à partir d'un navigateur Web

Pour pouvoir afficher ou contrôler une face-avant à distance via un navigateur Web, les clients qui n'ont 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 de l'ordinateur serveur crée un fichier HTML comprenant un tag <OBJECT> et <EMBED> 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 guident 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.

Si aucun autre client n'a le contrôle à ce moment et si aucune autre fenêtre de navigateur sur la même connexion n'a le contrôle, un message apparaît indiquant que vous avez le contrôle de la face-avant. En revanche, si un autre client contrôle le VI à ce moment-là, le serveur gardera votre demande dans une file d'attente jusqu'à ce que l'autre client abandonne le contrôle ou que la limite de temps de contrôle soit dépassée. Seul l'utilisateur de l'ordinateur serveur peut surveiller la liste des clients dans la file d'attente en sélectionnant **Outils»Gestionnaire de connexions de face-avant distante**.



Remarque National Instruments vous conseille d'utiliser Netscape 4.7 ou version ultérieure, ou encore Internet Explorer 5.5 Service Pack 2 ou version ultérieure pour afficher et contrôler des faces-avant distantes à l'aide d'un navigateur Web.

Fonctionnalités non supportées 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 dont l'interface utilisateur est complexe (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.

Évitez d'exporter des VIs qui comportent des boucles While, mais pas de fonction d'attente. Ces VIs empêchent les tâches en arrière-plan de s'exécuter en un laps de temps raisonnable, ce qui rend les faces-avant inertes et sans réponse lorsqu'elles sont affichées 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. De plus, le bouton Parcourir d'une commande chemin est également désactivé dans les faces-avant distantes.

Il se peut que des clients affichant 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.

Seul un contrôleur est en mesure d'afficher à distance la face-avant d'un VI chargé de manière dynamique qui est ouvert et exécuté à l'aide du VI Serveur ou la face-avant d'un sous-VI configuré pour afficher la face-avant lorsqu'il est appelé. Les clients qui ne contrôlent pas le VI ne peuvent pas afficher la face-avant.

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 soient moins performants 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.

Envoi de données par e-mail à partir de VIs

Utilisez les VIs E-mail SMTP pour envoyer des e-mails, y compris des données et des fichiers en pièces jointes, au moyen du protocole SMTP (Simple Mail Transfer Protocol). LabVIEW ne prend pas en charge l'authentification pour SMTP. Vous ne pouvez pas utiliser les VIs E-mail SMTP pour recevoir des informations. Les VIs E-mail SMTP utilisent le format MIME (Multipurpose Internet Mail Extensions) pour encoder les messages. Sous ce format, vous pouvez envoyer de multiples documents, notamment des fichiers de données binaires, dans un e-mail. Vous pouvez aussi décrire les propriétés de chaque pièce jointe, comme par exemple, le jeu de caractères utilisé par un document. Reportez-vous à l'*Aide LabVIEW* pour obtenir des informations sur l'envoi de données par e-mail à partir d'un VI.



Remarque Les VIs E-mail SMTP sont disponibles uniquement dans les systèmes de développement complet et professionnel (Full and Professional Development Systems).

Outre l'adresse e-mail du ou des destinataires, vous devez connaître l'adresse Web du serveur SMTP. Vous spécifiez un serveur de messagerie en câblant l'entrée **serveur mail** de chaque VI E-mail SMTP que vous utilisez. Le **serveur mail** doit être le nom d'hôte ou l'adresse IP d'un serveur externe qui peut gérer les demandes de l'ordinateur qui exécute les VIs E-mail SMTP. Si vous ne savez pas quel serveur mail utiliser, contactez votre administrateur réseau pour obtenir le nom d'un serveur valide. Après que vous avez spécifié un serveur mail valide, les VIs E-mail SMTP établissent une connexion avec le serveur et lui envoient les commandes qui décrivent les destinataires et le contenu de l'e-mail. Le serveur envoie le message à chaque destinataire ou le fait suivre à d'autres serveurs SMTP. Reportez-vous à `examples\comm\smtpex.llb` si vous voulez consulter des exemples d'utilisation des VIs E-mail SMTP.



Remarque Les VIs E-mail SMTP ne prennent pas en charge les caractères multi-octets, comme le japonais.

Sélection d'un jeu de caractères

Le paramètre d'entrée **jeu de caractères** des VIs E-mail SMTP spécifie le jeu de caractères devant être utilisé dans le texte de l'e-mail ou des pièces jointes. Un jeu de caractères décrit le mappage entre les caractères et leurs codes de caractères.

Un caractère est l'unité de base des langues écrites : une lettre, un nombre, un signe de ponctuation ou, dans certaines langues, un mot entier. Une modification apportée à une lettre, comme par exemple, une majuscule ou un accent, fait de cette lettre un autre caractère. Par exemple, O, o, Ö et Ô sont des caractères distincts.

Le code du caractère est un nombre qui représente un caractère. Comme les ordinateurs n'interprètent que les nombres, ils doivent associer un nombre à un caractère pour pouvoir traiter le caractère.

Un jeu de caractères est la relation entre les caractères et les nombres qui les représentent dans l'ordinateur. Par exemple, dans le jeu de caractères ASCII, les codes de caractères de A, B et C sont respectivement 65, 66 et 67.

Jeu de caractères US-ASCII

Un jeu de caractères couramment utilisé pour les messages e-mail est le jeu de caractères US-ASCII, ou ASCII. De nombreuses applications de messagerie utilisent ce jeu de caractères comme jeu par défaut et ne fonctionnent pas avec d'autres jeux de caractères. Le jeu de caractères ASCII renferme les lettres et la plupart des signes de ponctuation utilisés dans la langue anglaise, soit 128 caractères au total. La plupart des autres jeux de caractères sont des extensions de l'ASCII.

L'utilisation du jeu de caractères ASCII ne suffit pas dans tous les cas car de nombreuses langues nécessitent des caractères qui ne figurent pas dans le jeu ASCII. Par exemple, vous ne pouvez pas écrire le mot allemand Müller en utilisant les caractères ASCII car le caractère ü n'est pas défini dans le jeu ASCII.

Jeu de caractères ISO Latin-1

Du fait que de nombreuses langues nécessitent des caractères qui ne figurent pas au nombre des caractères ASCII, les pays qui utilisent ces langues ont créé de nouveaux jeux de caractères. La plupart de ces jeux de caractères contiennent les premiers 128 codes de caractères comme en ASCII et définissent les 128 codes de caractères suivants pour décrire les caractères spécifiques à une langue particulière. Certains de ces jeux de caractères utilisent différents codes de caractères pour décrire les mêmes caractères. Ceci entraîne parfois des problèmes si un jeu de caractères affiche du texte qui a été rédigé dans un autre jeu de caractères. Pour résoudre ce problème, utilisez un jeu de caractères standard. L'un de ces jeux de caractères standard couramment utilisé est l'ISO Latin-1, connu aussi sous l'appellation ISO-8859-1. Ce jeu de caractères décrit les caractères

utilisés dans la plupart des langues d'Europe occidentale et la plupart des applications de messagerie qui traitent ces langues comprennent ce jeu de caractères.

Jeu de caractères Mac OS

Apple Computer, Inc. a développé son propre jeu de caractères étendu avant que le jeu ISO Latin-1 n'ait été défini. Le jeu de caractères Mac OS est basé sur l'ASCII, mais utilise un autre jeu que l'ISO Latin-1 pour les 128 codes de caractères supérieurs. Pour cette raison, les e-mails qui contiennent des caractères accentués écrits dans le jeu de caractères Mac OS s'affichent de manière incorrecte dans les applications e-mail qui attendent du texte ISO Latin-1. Pour résoudre ce problème, les applications de messagerie Mac OS convertissent le texte en caractères ISO Latin-1 avant de l'envoyer. Lorsqu'une autre application de messagerie Mac OS reçoit du texte désigné comme utilisant le jeu de caractères ISO Latin-1, elle convertit le texte dans le jeu de caractères Mac OS.

Translitération

La translitération est le mappage de caractères à un autre jeu de caractères. Utilisez la translitération si vous envoyez un e-mail et que vous voulez changer le texte de l'e-mail en caractères correspondants d'un autre jeu de caractères. Vous pouvez utiliser les VIs E-mail SMTP pour spécifier les jeux de caractères qui mappent le texte à un autre jeu de caractères avant que le texte ne soit envoyé. Par exemple, si vous créez un message à l'aide des caractères ASCII standard et que vous spécifiez que le jeu de caractères est MacRoman, les VIs E-mail SMTP transposent le texte et l'envoient dans le jeu de caractères `iso-8859-1` (ISO Latin-1). Utilisez le paramètre d'entrée **translit** des VIs E-mail SMTP pour spécifier les translitérations qui peuvent être utilisées par ces VIs. Une translitération est définie par un jeu de caractères virtuel, un jeu de caractères cible et un fichier de translitération, ou mappage. Le fichier de translitération spécifie le mappage d'un caractère donné à un autre.

Le fichier de translitération est un fichier binaire de 256 octets, contenant 256 entrées. Chaque entrée du fichier correspond à un caractère particulier dans le jeu de caractères virtuel et contient le nouveau code de caractère provenant du jeu de caractères cible. Par exemple, si vous voulez mapper le caractère `a`, dont le code est `61`, au caractère `À`, dont le code est `41`, l'entrée dont l'indice est `61` dans le fichier de translitération doit contenir la valeur `41`. Si une entrée contient une valeur égale à son indice, le fichier de translitération ne modifie pas le caractère correspondant dans le jeu de

caractères virtuel. Par exemple, si l'entrée portant l'indice 61 dans le fichier de translitération contient la valeur 61, le fichier de translitération ne modifie pas le caractère.

Lorsque vous spécifiez un fichier de translitération comme jeu de caractères cible dans l'entrée **translit**, les correspondances de mappage sont appliquées dans l'ordre spécifié. Par exemple, si l'entrée de translitération est [MacRoman iso-8859-1 macroman.tr1, MacRomanUp MacRoman asciiup.tr1], le jeu de caractères MacRoman a été changé en jeu de caractères iso-8859-1 à l'aide de macroman.tr1 et MacRomanUp a ensuite été changé en MacRoman à l'aide de asciiup.tr1. Reportez-vous au répertoire vi.lib\Utility\SMTP pour consulter des exemples de fichiers .tr1.

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 Mac OS et Windows, vous devez utiliser un protocole qui fonctionne sur les deux plates-formes, comme 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 *Using LabVIEW with TCP/IP and UDP* pour obtenir de plus amples informations sur l'utilisation des protocoles TCP et UDP dans les applications de LabVIEW.

Utilisez le VI UDP Multicast - Ouvrir au lieu de la fonction UDP Ouvrir pour ouvrir des connexions capables de lire, d'écrire, ou de lire et d'écrire des données UDP en provenance et en direction d'une adresse IP multicast. Une adresse IP multicast définit un groupe multicast. Les adresses IP multicast sont dans la gamme 224.0.0.0 à 239.255.255.255. Lorsqu'un client veut se joindre à un groupe multicast, il souscrit à l'adresse IP multicast du groupe. Après que le client a souscrit à un groupe multicast, il reçoit les données envoyées à l'adresse IP multicast. Référez-vous à la note d'application *Using LabVIEW with TCP/IP and UDP* pour obtenir des informations complémentaires sur l'utilisation des interfaces de connexion multicast UDP.

Apple Events et PPC Toolbox (Mac OS)

La forme de communication réservée à Mac la plus courante est les Apple Events. Utilisez les Apple Events pour envoyer des messages de demande d'intervention ou de renvoi d'informations en provenance d'autres applications Mac OS.

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 Mac OS. La boîte à outils PPC est plus performante que les 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 obtenir 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 ils n'ont 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.

Connectivité Windows

LabVIEW permet l'accès à d'autres applications Windows au moyen des technologies .NET ou ActiveX. Vous pouvez utiliser LabVIEW comme client .NET pour accéder aux objets, aux propriétés et aux méthodes associées à des serveurs .NET. LabVIEW n'est pas un serveur .NET. D'autres applications ne peuvent pas communiquer directement avec LabVIEW par l'intermédiaire de .NET. Avec un VI activé .NET, vous pouvez vous connecter aux services et API de Windows. Le .NET framework comprend les services de composants COM+, ASP Web development framework et la prise en charge de protocoles de service Web, tels que SOAP, WSDL et UDDI. Le .NET framework constitue la base de programmation de l'environnement .NET que vous utilisez pour construire, déployer et exécuter des applications de type Web, des applications client intelligent et des services Web XML.

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 Windows. Vous pouvez utiliser LabVIEW comme client ActiveX pour accéder aux objets, aux propriétés, aux méthodes et aux événements associés à d'autres applications supportant ActiveX. LabVIEW peut aussi servir de serveur ActiveX, ce qui signifie que d'autres applications peuvent accéder à des objets, des propriétés et des méthodes de LabVIEW.

.NET fait référence à la technologie .NET de Microsoft. Vous devez installer .NET framework. Reportez-vous au site Web MSDN pour obtenir de plus amples informations sur .NET et sur l'installation du framework. *ActiveX* se rapporte aux technologies OLE et ActiveX de Microsoft. Reportez-vous à la documentation de Microsoft Developer's Network, *Inside OLE*, de Craig 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* et au site Web de National Instruments à l'adresse ni.com pour obtenir de plus amples informations sur l'utilisation de la technologie .NET et ActiveX.

Environnement .NET

La liste suivante fournit des informations de base sur les divers éléments qui constituent l'environnement .NET. Le but de ces informations est de vous aider à comprendre .NET, mais il n'est pas essentiel que vous appreniez ces informations pour pouvoir utiliser les composants .NET dans LabVIEW.

- **Common Language Runtime (CLR)** : ensemble de bibliothèques chargées des services d'exécution, comme l'intégration du langage, l'application des règles de sécurité, la mémoire, la collecte des données mises au rebut, la gestion des processus et la gestion des threads. Le CLR constitue la base de .NET et utilise le langage intermédiaire (IL) généré par tous les langages de programmation afin de faciliter la communication entre .NET et d'autres programmes.

Afin d'aider .NET à communiquer avec d'autres programmes, le CLR fournit un système de type de données qui couvre une vaste plage de langages de programmation et de systèmes d'exploitation. Les développeurs utilisent le CLR pour visualiser le système comme un ensemble de types de données et d'objets plutôt que comme un ensemble de dispositifs de mémoire et de threads. Le CLR nécessite des compilateurs et des éditeurs de liens pour générer des informations au format de métadonnées CLR IL. Sous un système Win32, tous les compilateurs de langages de programmation génèrent du code CLR IL au lieu du code assembleur.

- **Bibliothèques de classes** : ensemble de classes qui fournissent une fonctionnalité standard, comme les entrées et sorties, la manipulation de chaînes, la gestion de la sécurité, les communications réseau, la gestion des threads, la gestion du texte, les fonctionnalités de conception de l'interface utilisateur, etc. Ces classes contiennent les mêmes fonctions que celles qu'utilise le système Win32/COM. Dans le .NET framework, vous pouvez utiliser dans un langage .NET des classes créées dans un autre langage .NET.
- **Assemblées** : unité de déploiement semblable à une DLL, un OCX ou un exécutable pour une composante COM. Les assemblées sont des DLLs et des exécutables que vous construisez à l'aide du CLR .NET. Les assemblées consistent en un fichier unique ou en plusieurs fichiers. Une assemblée comprend un manifeste qui contient des informations sur le nom de l'assemblée, sur la version, des informations locales, des informations sur la sécurité fournies par l'éditeur, la liste des fichiers qui constituent l'assemblée, la liste des assemblées dépendantes, des ressources et des types de données exportés. Les assemblées à un seul fichier regroupent toutes ces données en un seul fichier, y compris le

manifeste et toutes les ressources nécessaires. Les assemblées à fichiers multiples peuvent avoir des ressources externes, comme des fichiers images, des icônes, des fichiers audio et autres, ou bien comporter un fichier pour le code principal et un autre pour les bibliothèques auxiliaires.

Les assemblées peuvent être publiques ou privées. .NET exige que les assemblées privées se trouvent dans le même répertoire que le répertoire d'application et que les assemblées publiques se trouvent dans une mémoire cache globale accessible par l'ensemble du système, désignée par GAC (Global Assembly Cache). En général, le développeur de l'application rédige des assemblées privées qui seront utilisées par cette application. Le développeur de l'assemblée décide également du mode de contrôle de la version. Le nom de l'assemblée est le nom de fichier, moins l'extension de fichier, le cas échéant, du fichier qui contient le manifeste.

- **Global Assembly Cache (GAC)** : la liste des assemblées publiques disponibles sur le système. Le GAC est semblable au registre utilisé par COM.

Fonctions et nœuds .NET

Utilisez les fonctions et nœuds LabVIEW suivants pour accéder aux objets, aux propriétés et aux méthodes associées à des serveurs .NET.

- Utilisez le Nœud du constructeur situé sur la palette **.NET** pour sélectionner le constructeur d'une classe .NET à partir d'une assemblée et créer une instance de cette classe à l'exécution. Quand vous placez ce nœud sur le diagramme, LabVIEW affiche la boîte de dialogue **Sélectionner un constructeur .NET**.
- Utilisez le Nœud de propriété, situé sur la palette **.NET**, pour obtenir (lire) et définir (écrire) les propriétés associées à une classe .NET.
- Utilisez le Nœud de méthode situé sur la palette **.NET** pour appeler les méthodes associées à une classe .NET.
- Utilisez la fonction Fermer la référence, située sur la palette **.NET**, pour fermer toutes les références aux objets .NET lorsque vous n'avez plus besoin de la connexion.
- Utilisez la fonction En une classe plus générique, située sur la palette **.NET**, pour adapter le type d'une référence .NET à sa classe de base.
- Utilisez la fonction En une classe plus spécifique, située sur la palette **.NET**, pour adapter le type d'une référence .NET à sa classe dérivée.

LabVIEW comme client .NET

Lorsque LabVIEW accède aux objets associés avec les assemblées .NET, il agit en tant que client .NET. L'utilisation de LabVIEW comme client .NET implique les trois étapes principales suivantes.

1. Créez un objet .NET au moyen d'un constructeur et établissez une référence à cet objet.
2. Câblez la référence de l'objet .NET à un nœud de propriété ou à un nœud de méthode et sélectionnez une propriété ou une méthode.
3. Fermez la référence à l'objet .NET pour fermer la connexion à l'objet.

Pour accéder à un objet .NET, utilisez le Nœud du constructeur sur le diagramme pour créer l'objet .NET exigé. Utilisez le Nœud du constructeur pour sélectionner une classe d'objet parmi une assemblée. Quand vous placez un Nœud du constructeur sur le diagramme, LabVIEW affiche la boîte de dialogue **Sélectionner un constructeur .NET**, dans laquelle sont répertoriées toutes les assemblées publiques dans la mémoire cache globale GAC. Si vous souhaitez sélectionner une assemblée privée, cliquez sur le bouton **Parcourir** de la boîte de dialogue et naviguez dans le système de fichier pour trouver une assemblée privée. Les assemblées .NET sont des types de fichier .dll et .exe. Après que vous avez sélectionné une assemblée privée, celle-ci apparaît dans le menu déroulant **Assemblée** de la boîte de dialogue **Sélectionner un constructeur .NET** à l'ouverture suivante de cette boîte de dialogue.

Lorsque vous sélectionnez une assemblée et une classe, les constructeurs de cette classe s'affichent dans la section **Constructeurs** de la boîte de dialogue **Sélectionner un constructeur .NET**. Sélectionnez le constructeur et cliquez sur le bouton **OK** pour fermer cette boîte de dialogue. LabVIEW affiche le nom de la classe que vous avez sélectionnée sur le Nœud du constructeur.
















Le Nœud du constructeur est semblable à la fonction Automation Ouvrir d'ActiveX, à cette exception près que le Nœud du constructeur peut spécifier des paramètres d'initialisation pour créer l'objet. Vous pouvez câbler la référence du serveur .NET entre le Nœud du constructeur et un nœud de propriété ou un nœud de méthode et sélectionner une propriété ou une méthode dans le menu local. Utilisez la fonction Fermez la référence pour fermer la référence à l'objet .NET.

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur la création d'objets .NET.

Mappage des types de données

LabVIEW convertit les types de données des paramètres des propriétés, méthodes et constructeurs .NET en types de données LabVIEW, de sorte que LabVIEW puisse lire et interpréter ces données. LabVIEW affiche les types de données qu'il ne peut convertir en tant que refnums .NET. La table suivante liste les types de données .NET et le type de données converti en type LabVIEW.

Tableau 19-1. Types de données .NET et LabVIEW

Types .NET	Types LabVIEW
System.Int32, System.UInt32, System.Int16, System.UInt16	 ,  ,  , 
System.String	
System.Boolean	
System.Byte, System.Char, System.UByte	 ,  , 
System.Single, System.Double, System.Decimal	 ,  , 
System.Array	Affiché sous forme de tableau du type correspondant
Enumeration	 (non consécutif)
DateTime	
Tout autre objet .NET	

Déploiement d'applications .NET

Une fois que vous avez créé un VI qui comprend des composantes .NET, vous pouvez construire ce VI pour en faire un exécutable, une LLB ou une DLL.

Déploiement d'un exécutable

Quand vous construisez des applications qui utilisent des objets .NET, vous devez copier les assemblées privées utilisées par les VIs dans le répertoire de l'application autonome et vous assurer que .NET framework est installé sur l'ordinateur cible.

Déploiement de VIs

Lorsque vous déployez des VIs, vous devez copier les assemblées privées utilisées par les VIs dans le répertoire du VI de niveau principal.

Assurez-vous que toutes les assemblées que les VIs utilisent se trouvent dans le même répertoire que celui du VI de niveau principal et que .NET framework est installé sur l'ordinateur cible.

Déploiement de DLLs

Quand vous déployez une DLL construite par LabVIEW, vous devez copier les assemblées privées utilisées par les VIs dans le répertoire de l'application qui utilise les DLLs construites par LabVIEW et vous assurer que .NET framework est installé sur l'ordinateur cible.

Configuration d'une application client .NET (avancée)

.NET fournit une capacité administrative aux applications par le biais de fichiers de configuration. Un fichier de configuration renferme le contenu XML et comporte en principe l'extension `.config`. Pour configurer une application client .NET LabVIEW, vous pouvez fournir un fichier de configuration pour un VI de niveau principal ou pour une application autonome. Donnez au fichier de configuration le nom de l'application suivi de l'extension `.config`, comme par exemple, `Mon App.vi.config` ou `MonApp.exe.config`.

Objets, propriétés, méthodes et événements ActiveX

Les applications pour lesquelles ActiveX est activé incluent des 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 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 17, [Contrôle des VIs par programmation](#), pour obtenir de plus amples informations sur les objets, les propriétés et les méthodes de l'environnement LabVIEW.

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 des données propres à l'événement.

Reportez-vous à la section *Événements ActiveX* de ce chapitre pour obtenir des informations sur l'utilisation du nœud Enregistrer un VI Callback d'événement pour gérer les événements ActiveX.

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 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 pour ouvrir un objet ActiveX.
- Utilisez le conteneur ActiveX pour accéder à un objet ActiveX et l'afficher sur la face-avant. Cliquez avec le bouton droit sur le conteneur, sélectionnez **Insérer un objet ActiveX** dans le menu local et sélectionnez ensuite l'objet auquel vous voulez accéder.
- Utilisez le Nœud de propriété pour obtenir (lire) et définir (écrire) les propriétés associées à un objet ActiveX.
- Utilisez le Nœud de méthode pour appeler les méthodes associées à l'objet ActiveX.
- Utilisez le nœud Enregistrer un VI Callback d'événement pour traiter des événements qui se produisent sur un objet ActiveX.
- Utilisez la commande et l'indicateur variant pour transmettre des données entre les commandes ActiveX. Reportez-vous à la section *Manipulation de données variant* du chapitre 5, *Construction du diagramme*, pour obtenir de plus amples informations 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 dans un conteneur.
- 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 Fermer la référence. 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 Automation personnalisées](#) de ce chapitre pour obtenir de plus amples informations sur les interfaces personnalisées.

Insertion d'un objet ActiveX sur la face-avant

Pour insérer un objet ActiveX sur la face-avant, cliquez avec le bouton droit sur le conteneur ActiveX, sélectionnez **Insérer un objet ActiveX** dans le menu local, puis sélectionnez la commande ou le document ActiveX que vous voulez insérer. 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 de plus amples informations 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 `Automation Ouvrir`, ni à fermer la référence à l'objet au moyen de la fonction `Automation 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 références.

Mode de conception des objets ActiveX

Cliquez avec le bouton droit sur le conteneur ActiveX et sélectionnez **Avancé»Mode de conception** dans le menu local pour afficher le conteneur en mode conception pendant que vous éditez le VI. En mode conception, aucun événement n'est généré et les procédures d'événement

ne s'exécutent pas. Le mode par défaut est le mode exécution, pour lequel vous interagissez avec l'objet comme un utilisateur le ferait.

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 voir et donner une valeur à toutes les propriétés associées à un contrôle ActiveX ou à un document dans un conteneur ActiveX. Pour accéder au navigateur de propriétés ActiveX, cliquez avec le bouton droit 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 est un moyen simple de fixer la valeur des propriétés d'un objet ActiveX de manière interactive, mais vous ne pouvez pas l'utiliser pour fixer ces valeurs par programme, et vous pouvez utiliser le navigateur de propriétés ActiveX uniquement sur des objets à l'intérieur d'un conteneur. Le navigateur de propriétés ActiveX n'est pas disponible en mode d'exécution, ni tant qu'un VI est en cours d'exécution.

Pages de propriétés ActiveX

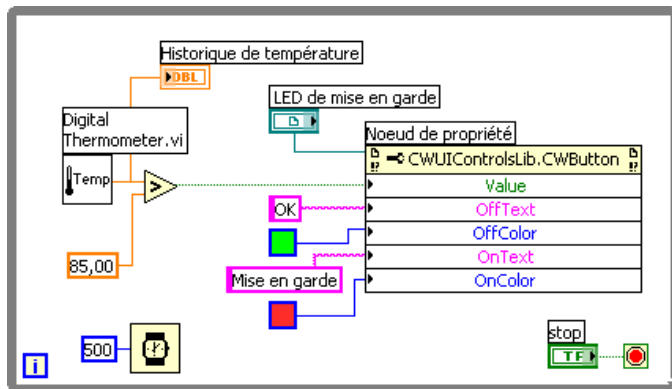
De nombreux objets ActiveX comprennent des pages de propriétés, qui organisent les propriétés associées aux objets sur des onglets séparés. Pour accéder aux pages de propriétés ActiveX, cliquez avec le bouton droit de la souris sur l'objet ou le document situé dans le conteneur de la face-avant et sélectionnez 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 objet 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. Les pages de propriétés ne sont pas non plus disponibles en mode Exécution ou tant que le VI est en cours d'exécution.

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 la Bibliothèque ActiveX de l'interface utilisateur Measurement Studio de National Instruments, quand la température atteint ou dépasse 85 degrés Fahrenheit.



Dans ce cas, l'objet CWButton se comporte comme un voyant (LED), change de couleur et affiche Mise en garde quand la température atteint la limite, ce qui correspond à l'état ON de l'objet CWButton.



Remarque Dans cet exemple, vous pourriez 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 de plus amples informations sur le navigateur de propriétés ActiveX et les pages de propriétés, respectivement.

LabVIEW en tant que serveur ActiveX

Les propriétés et méthodes de l'application LabVIEW, des VIs, et des commandes 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 d'un serveur ActiveX en utilisant LabVIEW, vous pouvez accéder aux interfaces personnalisées exposées par le serveur. Vous n'avez pas besoin d'utiliser IDispatch pour cela. Toutefois, le développeur du serveur ActiveX doit s'assurer que les paramètres des propriétés et des méthodes contenues dans ces interfaces personnalisées ont des types de données Automation (IDispatch). Le développeur du serveur doit effectuer cela afin d'exposer les interfaces multiples à partir d'un seul objet, plutôt qu'à partir de plusieurs objets. Vous pouvez toujours utiliser ces interfaces dans l'environnement LabVIEW. Reportez-vous à la documentation sur l'environnement de programmation de votre serveur pour obtenir des informations complémentaires sur l'accès aux interfaces personnalisées.

Utilisation de constantes pour définir des paramètres dans les VIs ActiveX

Certains paramètres des nœuds ActiveX dressent la 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 avec le bouton droit sur le paramètre du nœud acceptant les valeurs des données et sélectionnez **Créer une constante** dans le menu local. Les sélections disponibles dans la constante menu déroulant sont

fonction des refnums transmis au nœud. Les figures 19-1 et 19-2 présentent des exemples d'utilisation de la constante menu déroulant et de la constante numérique pour définir les valeurs des paramètres.

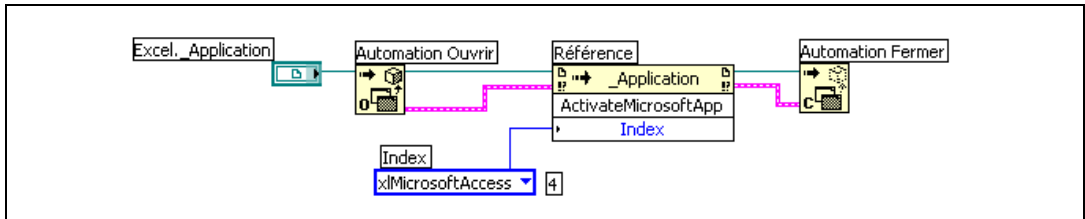


Figure 19-1. Paramétrage d'une valeur de donnée à l'aide d'une constante menu déroulant

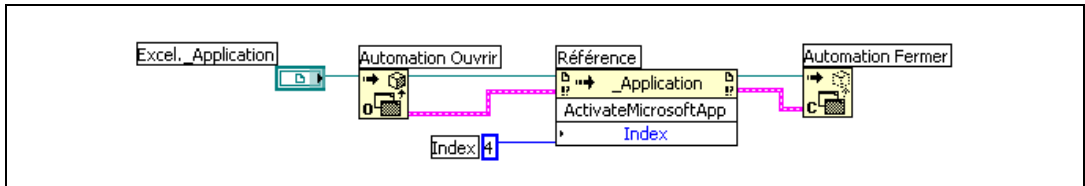


Figure 19-2. Paramétrage d'une valeur de donnée à l'aide d'une constante numérique

Les paramètres acceptant des valeurs de données ont une petite flèche à gauche de leur nom. Pour afficher la valeur des données correspondante, cliquez avec le bouton droit sur la constante menu déroulant et sélectionnez **Éléments visibles** > **Afficheur numérique** dans le menu local.

Les VIs des figures 19-1 et 19-2 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 19-1, 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 19-2.

Événements ActiveX

Pour utiliser les événements ActiveX dans une application, vous devez effectuer un enregistrement à l'événement et traiter celui-ci lorsqu'il se produit. L'enregistrement à un événement ActiveX est semblable à l'enregistrement dynamique à des événements, comme expliqué dans la section *Enregistrement dynamique d'événements* du chapitre 9, *Programmation événementielle*. Cependant, l'architecture d'un VI d'événement ActiveX est différente de celle d'un VI de gestion d'événement, comme décrit au chapitre 9, *Programmation événementielle*. Un VI d'événement ActiveX typique se compose des éléments suivants :

- L'objet ActiveX pour lequel vous voulez générer un événement.
- Le nœud Enregistrer un VI Callback d'événement pour spécifier le type d'événement que vous voulez générer et auquel vous voulez enregistrer l'objet.
- Le VI Callback qui contient le code que vous rédigez pour traiter l'événement que vous spécifiez.

Vous pouvez générer et gérer des événements sur des objets ActiveX dans un conteneur ou sur des objets ActiveX que vous spécifiez au moyen d'un refnum Automation. Par exemple, vous pouvez appeler une commande Windows de type arbre à partir d'un conteneur ActiveX et spécifier que vous voulez générer un événement Double Clic pour les éléments affichés dans la commande arbre.

Le nœud Enregistrer un VI Callback d'événement est un nœud extensible capable de gérer plusieurs événements, semblable au nœud Enregistrement à des événements.

Lorsque vous câblez une référence d'objet ActiveX à un nœud Enregistrer un VI Callback d'événement et que vous spécifiez l'événement que vous voulez générer pour cet objet, vous enregistrez l'objet à cet événement. Une fois que vous avez enregistré l'objet à l'événement, vous créez un VI Callback qui contient le code que vous rédigez en vue de gérer l'événement.

Gestion d'événements ActiveX

Vous devez créer un VI Callback pour gérer les événements de commandes ActiveX lorsque ces commandes génèrent les événements enregistrés. Le VI Callback s'exécute lorsque l'événement a lieu. Pour créer un VI Callback, effectuez un clic droit sur l'entrée **VI Ref** du nœud Enregistrer

un VI Callback d'événement et sélectionnez **Créer un VI Callback** dans le menu local. LabVIEW crée un VI réentrant qui contient les éléments suivants :

- **Données communes aux événements** contient les éléments suivants :
 - **Source** est une commande numérique qui spécifie la source de l'événement, comme LabVIEW ou ActiveX. Une valeur de 1 indique un événement ActiveX.
 - **Type** spécifie quel événement s'est produit. Il s'agit d'un type énumération pour les événement de l'interface utilisateur et d'un entier non signé 32 bits pour les événements ActiveX et les autres sources d'événements. Pour les événements ActiveX, le type d'événement représente le code de la méthode, ou ID, de l'événement enregistré.
 - **Heure** est l'horodatage en millisecondes spécifiant à quel moment l'événement a eu lieu.
- **CtlRef** est une référence au refnum Automation ou ActiveX sur lequel l'événement a eu lieu.
- **Données sur l'événement** est un cluster des paramètres spécifiques à l'événement que le VI Callback gère. LabVIEW détermine les **Données sur l'événement** appropriées lorsque vous sélectionnez un événement dans le nœud Enregistrer un VI Callback d'événement. Reportez-vous à la section *Événements de type filtre et notification* du chapitre 9, *Programmation événementielle*, pour obtenir de plus amples informations sur les événements de type notification et filtre.
- **Sortie de données sur l'événement** est un cluster des paramètres modifiables spécifiques à l'événement que le VI Callback gère. Cet élément est disponible uniquement pour les événements filtre.
- (Facultatif) **Paramètre utilisateur** correspond aux données que LabVIEW transmet à l'utilisateur par l'intermédiaire du VI Callback lorsque l'objet ActiveX génère l'événement.



Remarque Vous pouvez utiliser un VI existant comme VI callback à condition que le connecteur du VI que vous comptez utiliser corresponde au connecteur des données de l'événement. National Instruments vous recommande d'utiliser un VI réentrant car dans le cas contraire, LabVIEW ne peut pas l'appeler simultanément si des événements ActiveX se produisent plusieurs fois.

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 (CIN).

Reportez-vous à la section *Call Library Function Nodes and Code Interface Nodes* du chapitre 6, *LabVIEW Style Guide*, dans le manuel *LabVIEW Development Guidelines* pour obtenir de plus amples informations sur les considérations spécifiques aux plates-formes lors de l'appel de code externe. Reportez-vous au manuel *Using External Code in LabVIEW* pour obtenir de plus amples informations sur l'appel de code à partir de langages de programmation textuels.

Pour en savoir plus...

Reportez-vous à l'*Aide LabVIEW* pour obtenir de plus amples informations sur l'appel de code à partir 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 de câbler diverses fonctions arithmétiques 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 bénéficier d'une fonctionnalité supplémentaire, connectez-vous à l'application mathématique MATLAB.

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 d'expression et le nœud de script MATLAB pour effectuer des opérations mathématiques dans le diagramme.



Remarque Vous devez avoir MATLAB installé sur votre ordinateur pour pouvoir utiliser le nœud de script, car le nœud de script fait appel au serveur de script MATLAB pour exécuter les scripts MATLAB. Comme LabVIEW utilise la technologie ActiveX pour implémenter les nœuds de script, ces derniers ne sont disponibles que sous Windows. Les nœuds de script sont semblables à la boîte de calcul, mais ils permettent d'importer un script MATLAB existant au format ASCII et d'exécuter le script dans LabVIEW. Comme avec une boîte de calcul, vous pouvez transmettre des données en provenance et en direction 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.

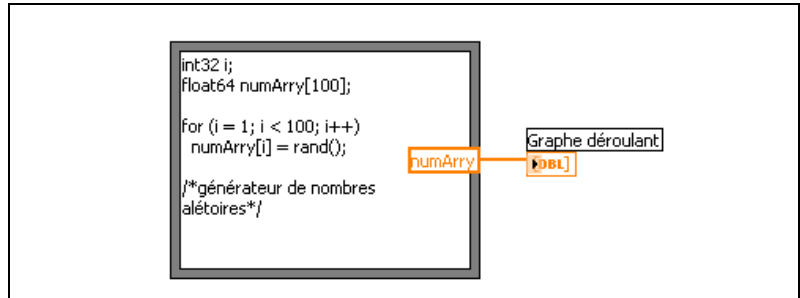
Les boîtes de calcul sont utiles pour résoudre des équations compliquées, comme les équations qui comportent de nombreuses variables, et pour utiliser un code textuel existant. Vous pouvez copier et coller le code textuel existant dans une boîte de calcul au lieu de le recréer graphiquement.

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 est une boîte redimensionnable semblable à la boucle For, la boucle While, la structure Condition, la structure Séquence empilée et la structure Séquence déroulée. 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 Create Function with Formula Node dans `examples\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 avec le bouton droit sur le bord de la boîte de calcul 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 avec le bouton droit sur le bord de la boîte de calcul 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

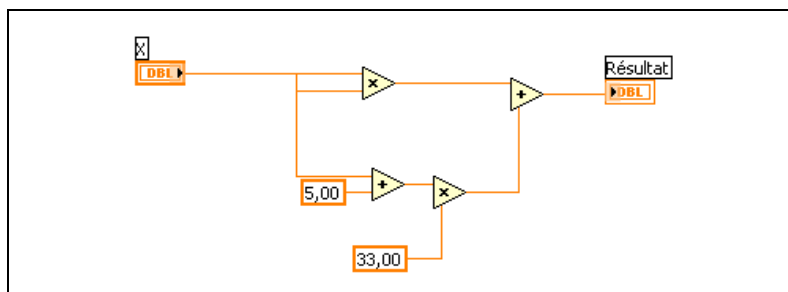
Utilisez les nœuds d'expression pour calculer des expressions ou des équations contenant une seule variable. Les nœuds d'expression sont utiles lorsqu'une équation n'a qu'une variable, mais qu'elle est malgré tout compliquée.

Les nœuds d'expression utilisent la valeur que vous transmettez au terminal d'entrée comme valeur de la variable. Le terminal de sortie renvoie la valeur du calcul.

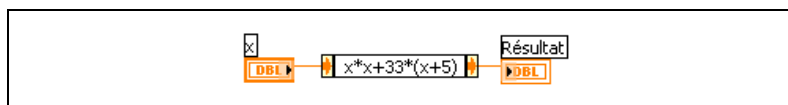
Par exemple, considérons l'équation simple suivante :

$$x \times x + 33 \times (x + 5)$$

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.

Nœuds de script MATLAB

Utilisez le nœud de script MATLAB pour créer, charger et éditer des scripts MATLAB sur le diagramme.









Vous devez avoir installé MATLAB pour pouvoir utiliser le nœud de script. Si vous avez déjà un script écrit 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 transmettre les valeurs 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 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. Le moteur du serveur de script est installé avec MATLAB.

En raison de la nature du langage de scripts MATLAB, le nœud de script ne peut pas déterminer le type de données des terminaux que vous avez créés. Vous devez affecter un type de données LabVIEW à chaque terminal du nœud de script. Tableau 21-1 présente les types de données LabVIEW et les types de données correspondantes dans MATLAB.

Tableau 21-1. Types de données LabVIEW et MATLAB

Type de données LabVIEW	Type de données MATLAB
	Réel
	Chaîne de caractères
	Chaîne de caractères
	Vecteur réel
	Matrice réelle
	Complexe
	Vecteur complexe
	Matrice complexe

Utilisez les fonctions de conversion ou les fonctions de conversion chaîne/tableau/chemin pour convertir un type de données LabVIEW en un type de données supporté par MATLAB.

Suggestions concernant la programmation des scripts MATLAB

Les techniques de programmation suivantes facilitent la mise au point d'un script :

- Écrivez le script et exécutez-le dans MATLAB pour le tester et le mettre au point avant de l'importer dans LabVIEW.
- Vérifiez les types de données. Quand vous créez une nouvelle entrée ou sortie, assurez-vous que le type de données du terminal est correct. Vous pouvez utiliser les fonctions Entrée d'erreur et Sortie d'erreur pour faire le suivi de cette opération.
- Créez des commandes et des indicateurs pour les entrées et les sorties afin de surveiller les valeurs que le script transmet entre LabVIEW et MATLAB. Ceci vous permet de localiser, si besoin est, l'endroit où un nœud de script a calculé une valeur de manière incorrecte.
- Tirez parti des paramètres de détection 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 afficher les informations sur les erreurs pendant l'exécution. Les boîtes de calcul affichent aussi les erreurs à la compilation.



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, Mac OS 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, *Configuring Measurement Hardware*, du *LabVIEW Measurements Manual* pour obtenir de plus amples informations sur la configuration de votre matériel.

Le répertoire LabVIEW contient les ensembles suivants une fois l'installation terminée.

Bibliothèques

- `user.lib` : répertoire dans lequel vous enregistrez les commandes et les VIs que vous créez. LabVIEW affiche les commandes sur les palettes **Commandes utilisateur** et les VIs sur les palettes **Bibliothèques utilisateur**. Ce répertoire ne change pas si vous mettez LabVIEW à niveau ou si vous le désinstallez. Reportez-vous à la section *Ajout de VIs et de commandes aux sous-palettes Utilisateur et Drivers d'instruments* du chapitre 3, *Environnement LabVIEW*, pour obtenir de plus amples informations sur l'enregistrement de fichiers dans le répertoire `user.lib`.
- `vi.lib` : contient les bibliothèques de VIs intégrés, que LabVIEW affiche dans des groupes apparentés dans la palette de **Fonctions**. N'enregistrez pas de fichiers dans le répertoire `vi.lib` parce que LabVIEW écrase les fichiers dans ce répertoire lorsque vous effectuez une mise à jour ou une réinstallation.
- `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 et que vous les placez dans ce répertoire, LabVIEW les ajoute à la palette **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 ou une réinstallation.
- `project` : contient des fichiers qui deviennent des éléments du menu **Outils** de LabVIEW.
- `templates` : contient des modèles pour des VIs ordinaires.
- `www` : emplacement des fichiers HTML auxquels vous pouvez accéder par le serveur Web.

Exercices et exemples

- `examples` : contient des VIs d'exemple. Sélectionnez **Aide» Recherche d'exemples** pour parcourir ou rechercher les 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...**

Mac OS

En plus des répertoires mentionnés précédemment, les utilisateurs Mac 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` : Toutes les commandes ou VIs que vous souhaitez afficher dans les palettes **Commandes utilisateur** ou **Bibliothèques utilisateur**. Reportez-vous à la section *Ajout de VIs et de commandes aux sous-palettes Utilisateur et Drivers d'instruments* du chapitre 3, *Environnement LabVIEW*, pour obtenir de plus amples informations sur l'enregistrement de fichiers dans le répertoire `user.lib`.



Remarque N'enregistrez dans le répertoire `user.lib` que des sous-VIs portables, sans modification, d'un projet à l'autre. Les chemins vers des VIs de `user.lib` sont relatifs au répertoire `labview`. Les chemins vers des sous-VIs que vous avez enregistré 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 driver d'instrument que vous voulez afficher dans la palette **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**.
- LabVIEW Data : Les fichiers de données générés par LabVIEW, comme les fichiers `.lvn` ou `.txt`.

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 la création de répertoires.

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 la création et l'utilisation des fonctions polymorphes.

For more information...

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 convertissent ces entiers en nombres à virgule flottante double précision avant d'effectuer le calcul.

Pour représenter une quantité scalaire de type flottant, il est recommandé d'utiliser des nombres à virgule flottante double précision. Les nombres à virgule flottante simple précision ne font gagner que peu de temps

d'exécution, sinon pas du tout, et ils sont davantage sujets au débordement. Les bibliothèques d'analyse, par exemple, utilisent des nombres à virgule flottante 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 :

- **Nombre entier signé ou non signé en nombre flottant** : la conversion est exacte, excepté pour les entiers 32 bits en nombres flottants à 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 entiers, comme le terminal d'itération d'une boucle For, arrondissent les flottants. LabVIEW arrondit la partie fractionnaire de 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 copie uniquement les bits de poids faible 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.

La figure B-1 illustre les possibilités de combinaisons polymorphes de la fonction Additionner.

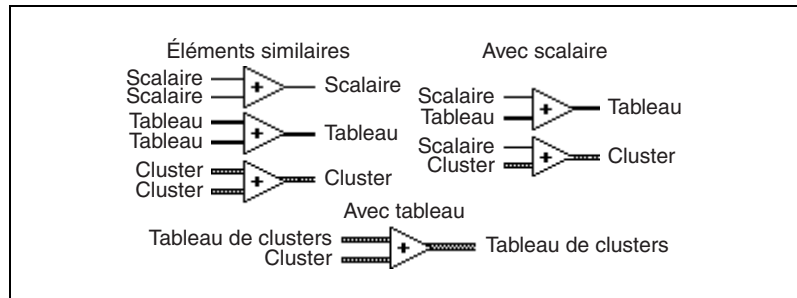


Figure B-1. Combinaisons polymorphes de la fonction Additionner

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. La figure B-2 présente des combinaisons de valeurs booléennes pour la fonction ET.

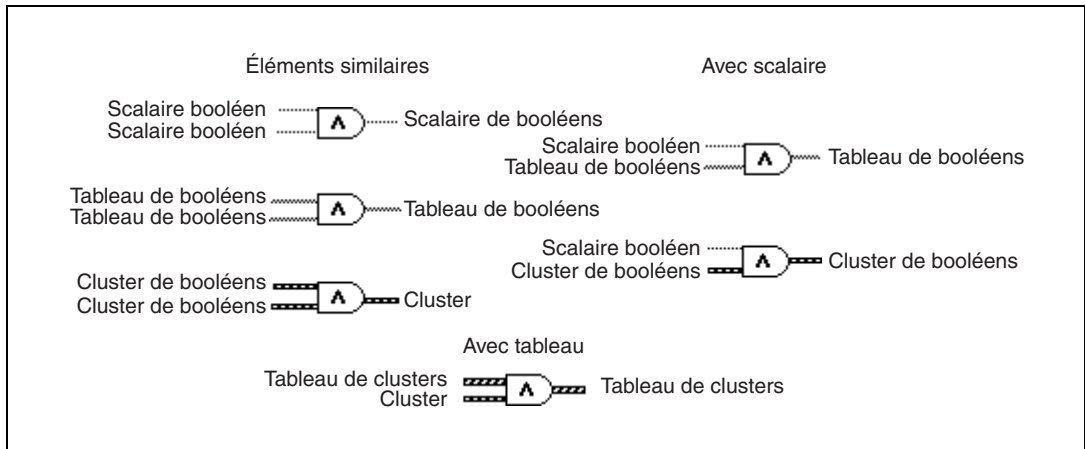


Figure B-2. Combinaisons de valeurs booléennes pour la fonction ET

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 Égax ?, Non Égax ? 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. Si vous voulez que la fonction renvoie une seule valeur booléenne, cliquez avec le bouton droit sur la fonction et sélectionnez **Mode de comparaison»Comparer des ensembles** dans le menu local. Reportez-vous à la section *Comparaison des tableaux et clusters* de l'annexe C, *Fonctions de comparaison*, pour obtenir de plus amples informations sur la manière dont cette fonction compare des ensembles.

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 cette annexe 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 pour comparer des valeurs booléennes, des chaînes, 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.

For more information...

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. Dans le mode Comparer des ensembles, si vous comparez deux tableaux ou clusters, la fonction renvoie une seule valeur booléenne. Dans le mode Comparer des éléments, la fonction compare individuellement les éléments et renvoie un tableau 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 ASCII.

Cliquez sur une fonction de comparaison et sélectionnez **Mode de comparaison»Comparer des éléments** ou **Mode de comparaison»Comparer des ensembles** dans le menu local pour changer le mode de la fonction. 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 fonctionnent uniquement en mode Comparer des ensembles 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 renvoient un tableau de valeurs booléennes de même dimension 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 un seul résultat booléen après comparaison des éléments d'un tableau. Les fonctions de comparaison considèrent les éléments qui figurent plus loin dans le tableau comme secondaires par rapport à ceux qui figurent plus haut dans le tableau. 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, la fonction s'arrête : elle renvoie le résultat de cette comparaison.
- Si les éléments sont égaux, la fonction de comparaison traite la paire de valeurs suivante, jusqu'à ce qu'elle 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 des tableaux comporte des éléments supplémentaires, le tableau le plus long est considéré comme étant supérieur à l'autre. Par exemple, une fonction de comparaison 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. Par exemple, vous pouvez comparer un cluster composé d'un DBL et d'une chaîne 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 un booléen unique. Cette fonction compare les éléments correspondants jusqu'à ce qu'elle trouve une différence, qui détermine le résultat. La fonction ne considère les deux clusters 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, si vous comparez un cluster contenant deux chaînes, une pour le nom de famille suivie d'une autre pour le prénom, la fonction compare les champs correspondant au prénom seulement si les champs correspondant au nom de famille concordent.

Support technique et services professionnels

Les sections suivantes du site National Instruments ni.com vous permettront d'obtenir des informations sur le support technique et les services professionnels :

- **Support** : les ressources de support technique sont les suivantes :
 - **Documents du support technique** : pour prendre connaissance des solutions ou recevoir des réponses immédiates à vos problèmes, consultez notre bibliothèque de ressources du support technique disponible en anglais, japonais et espagnol sur ni.com/support. Ces ressources couvrent la plupart des produits, gratuitement pour les utilisateurs enregistrés et incluent des mises à jour de nos logiciels, une base de données de connaissances (KnowledgeBase), les manuels des produits, des assistants de dépannage, des normes de conformité, des exemples et leur code source, des tutoriaux et notes d'applications, des drivers d'instruments, des forums de discussion, un glossaire sur la mesure et l'automatisation, etc.
 - **Services du support technique** : pour parler à un ingénieur du support technique, veuillez contacter votre filiale locale en consultant le site ni.com/niglobal. Pour interroger le support technique en France, visitez la page ni.com/france/support. Enfin, nous disposons d'un service de support technique en anglais à l'adresse ni.com/support. Ce système vous permet de formuler votre question et vous met en contact avec un expert, soit au support technique, soit sur un forum de discussion, soit par e-mail.
- **Formation** : le site ni.com/training vous permet de consulter des tutoriaux, vidéos et CD interactifs d'auto-formation. Vous pouvez également vous inscrire à des cours de formation dispensés par des instructeurs dans de nombreuses villes.
- **Intégration** : si votre temps est compté ou vos ressources humaines limitées, ou si vous avez besoin de compétences pointues, les membres du Programme Alliance NI sont là pour vous aider. Pour en savoir plus, appelez votre filiale locale ou visitez le site ni.com/alliance.

Si le site ni.com n'a pas complètement répondu à vos attentes, n'hésitez pas à contacter votre filiale locale. Les numéros de téléphone des filiales figurent au début de ce manuel. Vous pouvez également visiter la page ni.com/niglobal pour obtenir les coordonnées des filiales internationales, consulter leur site Web respectif et vous tenir au courant des événements régionaux.

Glossaire

Symboles	Préfixe	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 dimensions.

3D À trois dimensions.

A

A Ampères.

À deux dimensions Objet ayant deux dimensions, comme c'est le cas d'un tableau comprenant plusieurs lignes et plusieurs colonnes.

À une dimension Objet ayant une dimension, comme c'est le cas d'un tableau qui ne comporte qu'une ligne d'éléments.

Acquisition de données (DAQ)

1. Acquisition et mesure de signaux électriques analogiques ou numériques provenant de capteurs, de transducteurs et de sondes ou d'appareils de test.
2. Génération de signaux électriques analogiques ou numériques.

animation de l'exécution Technique de mise au point qui consiste à animer l'exécution d'un VI pour illustrer la manière dont les données circulent dans le VI.

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 analogiques et numériques DAQ. Disponible dans le Voisinage de données du Measurement & Automation Explorer (Windows) ou DAQ Channel Wizard (Mac OS).
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 1D, les 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	<i>Voir</i> bibliothèque de VIs.
bibliothèque de VIs	Fichier spécial contenant une collection de VIs apparentés destinés à une utilisation particulière.
bibliothèque partagée	Un fichier contenant un module programme exécutable que d'autres programmes peuvent utiliser pour exécuter une fonction. Les bibliothèques partagées sont utiles lorsque vous souhaitez partager une fonctionnalité des VIs que vous créez avec d'autres développeurs.
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 en 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 en 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 d'un fil de liaison contenant tous les segments de fil de liaison d'une jonction à une autre, d'un terminal à une jonction ou d'un terminal à un terminal, si ces derniers ne sont séparés par aucune jonction.
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 déroulée ou empilée.
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 des options multiples que vous pouvez définir. Vous pouvez sélectionner plus d'une case à cocher.
cercles ou poignées de redimensionnement	Cercles ou poignées de dimensionnement qui apparaissent sur les bords d'un objet pour indiquer les points où vous pouvez redimensionner l'objet.
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	Catégorie contenant des propriétés, des méthodes et des événements. Les classes sont organisées en une hiérarchie selon laquelle chaque classe hérite des propriétés et méthodes associées à la classe du niveau précédent.
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 son nouvel emplacement. (Mac OS) 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)	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 numériques.	Objets de la face-avant utilisés pour manipuler et afficher des données booléennes (VRAI ou FAUX).

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	<ol style="list-style-type: none"> Partie du VI ou du nœud de fonction qui contient les terminaux d'entrée et de sortie. Les données sont transmises en provenance ou à destination d'un nœud par le biais d'un connecteur. Zone dans le coin droit supérieur de la face-avant ou du diagramme qui affiche le modèle des terminaux du VI. Il définit les entrées et les sorties que vous pouvez câbler à un VI.
constante	Voir constante universelle <i>et</i> constante définie par l'utilisateur.
constante définie par l'utilisateur	Objet du diagramme qui émet une valeur que vous définissez.
constante universelle	Objet de diagramme non éditable qui correspond à un caractère ASCII particulier ou 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 actuelle 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

D	Delta ; Différence. Δx correspond à la valeur par laquelle x passe d'un indice au suivant.
DAQ	Voir Acquisition de données (DAQ) .
déclenchement externe	Impulsion en tension à partir d'une source externe qui déclenche un événement, comme une conversion analogique/numérique par exemple.

définition de type	Copie principale d'un objet personnalisé que plusieurs VIs peuvent utiliser.
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. <i>Voir aussi</i> dépendance de données artificielle.
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.
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	Représente des lectures multiples effectuées sur un signal à une fréquence d'échantillonnage particulière.
driver	Logiciel exclusivement conçu pour un périphérique ou un type de périphérique et qui comprend le jeu de commandes accepté par ce périphérique.
driver d'instrument	VI qui contrôle un instrument programmable.

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, de graphe déroulant et de certaines commandes et indicateurs numériques qui contient une série de repères ou de points situés à des intervalles connus pour représenter les unités de mesure.
enregistrement dans un fichier journal	Habituellement utilisé pour acquérir des données et les stocker simultanément dans un fichier sur le disque. Les VIs et les fonctions d'E/S sur fichiers de LabVIEW peuvent enregistrer des données dans un journal.
entier	Tous les nombres entiers naturels, leur opposé, ou zéro.
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.
événement de type filtre	Permettent de contrôler la manière dont l'interface utilisateur se comporte.
événements de type notification	Préviennent LabVIEW qu'une action utilisateur a déjà eu lieu, comme le changement de la valeur d'une commande.

F

face-avant	Interface utilisateur interactive d'un VI. L'apparence de la face-avant imite 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 titre d'une fenêtre active est mise en évidence. Pour activer une fenêtre, cliquez dessus ou sélectionnez-la dans le menu Fenêtres .
fenêtre Aide contextuelle	Fenêtre qui affiche des informations de base sur les objets de LabVIEW lorsque vous déplacez le curseur au-dessus de chaque objet. Les objets comportant des informations d'aide contextuelle comprennent les VIs, les fonctions, les constantes, les structures, les palettes, les propriétés, les méthodes, les événements et les composantes de boîtes de dialogue.
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, les icônes et le connecteur.
fichier journal	Fichier qui stocke les données comme séquence d'enregistrements de type de données simples et arbitraires que vous spécifiez lorsque vous créez le fichier. Tous les enregistrements dans 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 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 de données entre des 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 constitué de nœuds exécutables qui s'exécutent seulement s'ils reçoivent toutes les données d'entrée nécessaires et qui produisent automatiquement une sortie lors de leur exécution. LabVIEW est un système de 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. 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 amovible des commandes et indicateurs à glissière.
GPIB	<i>Voir</i> General Purpose Interface Bus.
graphe	Affichage 2D d'au moins une courbe. Un graphe reçoit des données et les trace en bloc.
graphe à balayage	Indicateur numérique qui fonctionne sur le modèle d'un oscilloscope. Similaire à un oscillographe, sauf que les données anciennes sont séparées des données nouvelles par une ligne qui traverse l'écran d'affichage.
graphe déroulant	<ol style="list-style-type: none"> 1. Affichage 2D d'au moins une courbe dans laquelle l'affichage conserve l'historique des 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. <i>Voir aussi</i> graphe déroulant et graphe à balayage. 2. Indicateur qui trace les points de données à une certaine fréquence. 3. Indicateur de traçage numérique sur le modèle d'un enregistreur papier de graphe déroulant, qui défile au fur et à mesure qu'il trace des données.

groupe Ensemble des voies ou des 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 fois.

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 renvoie une mise en garde. LabVIEW ne redémarre pas la tâche une fois que vous avez reconfiguré le groupe.

H

hexa Hexadécimal. Système de nombres de base 16.

I

icône Représentation graphique d'un nœud du diagramme.

IEEE Institute for Electrical and Electronic Engineers.

image Série d'instructions graphiques qu'un indicateur d'images utilise pour créer une image.

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.

Indicateurs et commandes numériques Objets de la face-avant utilisés pour manipuler et afficher des données numériques.

Inf Valeur d'affichage numérique pour une représentation de l'infini par nombres à virgule flottante.

info-bulle Petites bannières de texte jaunes qui identifient le nom du terminal et facilitent l'identification des terminaux pour le câblage.

inplace La condition dans laquelle plusieurs terminaux, tels que des terminaux d'E/S ou des registres à décalage, utilisent le même espace mémoire.

instrument virtuel (VI) Programme de LabVIEW modélisant l'apparence et le fonctionnement d'un instrument réel.

IP Protocole Internet (Internet Protocol).

J

jonction d'un fil de liaison Point ou trois segments de fils de liaison ou plus se rejoignent.

L

LabVIEW Laboratory Virtual Instrument Engineering Workbench. LabVIEW est un langage de programmation graphique qui utilise des icônes au lieu de 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 ou un graphe déroulant pour afficher les noms et les styles de ses 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 Applications créées au moyen du système de développement LabVIEW et exécutées dans l'environnement du système d'exécution LabVIEW.

M

matrice Un tableau rectangulaire de nombres ou d'éléments mathématiques qui représente les coefficients dans un système d'équations linéaires.

Measurement & Automation Explorer Environnement standard de configuration et de diagnostic du matériel de National Instruments pour Windows.

menu local	Menu auquel vous accédez en cliquant sur l'objet avec le bouton droit de la souris. Les éléments de menu sont spécifiques à l'objet.
menus déroulants	Menus auxquels vous accédez à partir d'une barre de menus. Les éléments de menu déroulant sont habituellement des menus de nature générale.
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.

N

NaN	Valeur d'afficheur numérique pour une représentation en flottants de <i>pas un nombre</i> . Généralement le résultat d'une opération non définie, comme un enregistrement(-1).
NI-DAQ	Logiciels pilotes inclus avec tous les appareils de mesure de National Instruments. NI-DAQ est une vaste bibliothèque de VIs et de fonctions que vous pouvez appeler à partir d'un environnement de développement d'application (ADE), comme LabVIEW, afin de programmer toutes les fonctionnalités d'un périphérique de mesure National Instruments, notamment la configuration, l'acquisition de données, la génération de données par ce périphérique, ainsi que l'envoi de données à ce périphérique.
NI-DAQmx	Le tout dernier driver NI-DAQ comportant de nouveaux VIs, fonctions et outils de développement pour le contrôle des périphériques de mesure. Parmi les avantages qu'offre NI-DAQmx comparé aux versions antérieures de NI-DAQ figurent : l'Assistant DAQ pour la configuration des tâches de mesure, des voies et des échelles, de meilleures performances, une fonctionnalité étendue et une interface de programmation d'applications (API) plus conviviale.
nœud	Élément d'exécution d'un programme. Les nœuds sont analogues aux déclarations, opérateurs, fonctions et sous-programmes des langages textuels. 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 géré par le cas que vous avez configuré. Si vous configurez une condition unique pour gérer plusieurs événements, seules les données communes à tous les types d'événements gérés seront disponibles.
Nœud de propriété	Définit ou obtient 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 servant à définir les chemins des données entre les 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 une 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 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.
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.
périphérique	Instrument ou contrôleur qui est adressable en tant qu'entité unique et qui contrôle ou surveille les points réels d'E/S. Un périphérique est le plus souvent connecté à un ordinateur hôte par l'intermédiaire d'un réseau de communication.
pixel	La plus petite unité d'une image numérisée.
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	Apparaît sur les nœuds du diagramme pour vous avertir que vous avez câblé ensemble des données de différents types de données numériques. Apparaît aussi lorsque vous câblez n'importe quel type de donnée à un type de donnée variant.
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 informatique modulaire.

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</i> refnum.
registre à décalage	Mécanisme optionnel dans les structures de type boucle permettant de transmettre la valeur d'une variable entre des itérations de boucle successives. 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 données numériques, qui peut être représenté par des entiers signés ou non signés 8, 16 ou 32 bits, de même que des nombres de types flottant simple précision, flottant double précision ou flottant précision étendue.

S

scalaire	Nombre qu'un point peut représenter sur une échelle. 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	Une seule partie horizontale ou verticale du fil de liaison.

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 contrôle de programme, tel qu'une structure Séquence déroulée, structure Séquence empilée, structure Condition, boucle For et 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 chaîne de caractères.
structure Séquence	Voir structure Séquence déroulée <i>ou</i> structure Séquence empilée.
structure Séquence déroulée	Structure de commande de programme qui exécute ses sous-diagrammes dans l'ordre numérique. Utilisez cette structure pour forcer les nœuds qui n'ont pas de données dépendantes à s'exécuter dans l'ordre que vous avez défini. La structure Séquence déroulée affiche toutes les étapes à la fois et exécute les étapes depuis la gauche vers la droite jusqu'à ce que la dernière étape s'exécute.
structure Séquence empilée	Structure de commande de programme qui exécute ses sous-diagrammes dans l'ordre numérique. Utilisez cette structure pour forcer les nœuds qui n'ont pas de données dépendantes à s'exécuter dans l'ordre que vous avez défini. La structure Séquence empilée affiche chaque étape en ne permettant de voir qu'une seule étape à la fois et exécute les étapes dans l'ordre jusqu'à ce que la dernière étape s'exécute.
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 ne 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/IP	Transmission Control Protocol/Internet Protocol. Un format standard de transmission de données en packets d'un ordinateur à un autre. Les deux composantes du TCP/IP sont le TCP, qui concerne la construction de paquets de données, et le IP, qui les achemine d'un ordinateur à un autre.
temps du système LabVIEW	Date et heure utilisées par LabVIEW comme référence de temps absolue. Le temps système de LabVIEW est défini à partir du 1er janvier 1904 à minuit, temps universel.
terminal	Objet ou région sur un nœud à travers lequel les données transitent.
terminal conditionnel	Terminal de boucle While contenant une valeur booléenne qui détermine si le VI effectue une autre itération.
terminal d'itération	Terminal d'une boucle For ou d'une boucle While qui contient le nombre courant d'itérations effectuées.
terminaux de décompte	Terminal d'une boucle For dont la valeur détermine le nombre de fois qu'une boucle For exécute son sous-diagramme.
tunnel	Terminal d'entrée ou de sortie de données sur une structure.
type de données	Format pour informations. Dans LabVIEW, les types de données acceptés 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.
URL	Uniform resource locator. Une adresse logique qui identifie une ressource sur un serveur, généralement sur le Web. Par exemple, <code>http://www.ni.com/</code> est l'adresse URL pour le site Web de National Instruments.
utilitaire de configuration	Se réfère à Measurement & Automation Explorer sous Windows et à l'utilitaire de configuration NI-DAQ sur Mac OS.

V

valeur par défaut	Valeur prédéfinie. De nombreuses entrées du VI utilisent une valeur par défaut si vous ne spécifiez pas de valeur.
variable globale	Accède à plusieurs VIs dans un diagramme et fait passer 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 étapes d'une structure Séquence empilée.
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 dans le bouton Exécution interrompue.
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 à une seule interface pour contrôler le bus GPIB, VXI, RS-232 et d'autres types d'instruments.

VISA	<i>Voir</i> Virtual Instrument Software Architecture .
voie	<p>1. Physique : terminal ou broche à l'emplacement duquel vous pouvez mesurer ou générer un signal analogique ou numérique. Une seule voie physique peut comprendre plus d'un terminal, comme dans le cas d'une voie d'entrée analogique différentielle ou d'un port numérique de huit lignes. Un compteur peut aussi être une voie physique, bien que le nom du compteur ne soit pas le nom du terminal où le compteur mesure ou génère le signal numérique.</p> <p>2. Virtuel : ensemble de paramètres de propriété pouvant inclure un nom, une voie physique, des connexions de terminal d'entrée, le type de mesure ou de génération et les informations de mise à l'échelle. Vous pouvez définir des voies virtuelles NI-DAQmx en dehors d'une tâche (globales) ou au sein d'une tâche (locales). La configuration de voies virtuelles est facultative dans NI-DAQ traditionnel et ses versions antérieures, mais fait partie intégrale de toute mesure réalisée dans NI-DAQmx. Dans NI-DAQ traditionnel, vous configurez des voies virtuelles dans MAX. Dans NI-DAQmx, vous pouvez configurer des voies virtuelles soit dans MAX, soit dans votre programme et configurer ces voies dans le cadre d'une tâche ou séparément.</p> <p>3. Switch : une voie switch représente un point quelconque de connexion sur un switch. Elle peut se composer d'un ou plusieurs fils de liaison (normalement un, deux ou quatre), suivant la topologie du switch. Une voie virtuelle ne peut pas être créée avec une voie switch. Les voies switch peuvent être utilisées uniquement dans les fonctions et VIs Switch NI-DAQmx.</p>
VXI	Extensions VME pour l'Instrumentation (bus).

Index

Symboles

.NET

déploiement d'applications, 19-5

.NET framework, 19-2

A

abaques de Smith, 13-3

acheminement particulier. *Voir l'Aide LabVIEW.*

acquisition

d'un sous-ensemble de données
numériques, 4-26

acquisition de données. *Voir DAQ.*

activation par simple clic. *Voir l'Aide LabVIEW.*

ActiveX, 19-1

accès aux applications pour lesquelles
ActiveX est activé, 19-8

affichage des propriétés, 19-10

appel de méthodes. *Voir l'Aide LabVIEW.*

clients, 19-8

commandes, 19-7

constantes pour définir des
paramètres, 19-12

construction des sous-palettes, 3-8

conteneurs, 19-7

définition de paramètres à l'aide de
constantes, 19-12

définition des propriétés, 19-10

définition des propriétés par
programmation, 19-11

événements, 19-7, 19-14

faces-avant distantes, 18-17

fonctions, 19-7

gestion des événements, 19-14

indicateurs, 19-7

insertion d'objets sur la face-avant, 19-9

interfaces personnalisées, 19-12

mise en réseau et, 18-1

mode de conception, 19-9

Navigateur de propriétés, 19-10

Nœud de propriété, 19-11

objets, 19-6

pages de propriétés, 19-10

pour les nœuds de script en cours
d'exécution, 21-1

propriétés, 19-6

sélection d'interfaces personnalisées, 19-9

serveurs, 19-12

VI Callback, 19-15

VI Serveur, 17-1

VI, 19-7

affichage

chaîne d'appelants, 6-11

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, 18-13

info-bulles. *Voir l'Aide LabVIEW.*

prises en garde, 6-3

objets masqués de la face-avant. *Voir l'Aide
LabVIEW.*

terminaux, 5-2

aide

Voir aussi fenêtre Aide contextuelle.

services professionnels, D-1

support technique, D-1

aiguilles

ajout, 4-12

ajout

ajout d'espace à la face-avant, 4-10

commandes à des bibliothèques, 3-7

d'instances à des VIs polymorphes. *Voir l'Aide LabVIEW.*

graphique à l'icône du VI. *Voir l'Aide LabVIEW.*

répertoires au chemin de recherche du VI. *Voir l'Aide LabVIEW.*

terminaux vers fonctions, 5-12

VIs à des bibliothèque, 3-7

alignement des objets, 4-6
Voir aussi l'Aide LabVIEW.

alignement lié à la grille, 4-6

animation de l'exécution
mise au point des VIs, 6-6

annotations, 4-30
édition. *Voir l'Aide LabVIEW.*

annulation d'erreurs existantes. *Voir l'Aide LabVIEW.*

annuler options, 3-9

appel à distance des VIs, 17-1

appel de code provenant de langages de programmation textuels, 20-1

appel de VIs de façon dynamique, 17-8

appelants
affichage, 6-11
chaîne d', 6-11

Appeler une fonction d'une DLL, 20-1

appellation
commandes, 7-12
VIs, 7-15

Apple events, 18-22

Application Builder. *Voir applications autonomes.*

applications
construction autonome, 7-16
distribution des VIs, 7-16
construire un VI Serveur, 17-2

applications autonomes
construction, 7-16
distribution des VIs, 7-16
.NET, 19-5

arithmétiques. *Voir équations.*

ASCII
utilisation du jeu de caractères, 18-19

Assistant des sondes personnalisées, 6-9

attribution
mots de passe pour les diagrammes, 7-16

attributs
données variant, 5-26

audio, 13-8

auto-indexation
boucles For, 8-5
boucles While, 8-6
données par défaut, 6-13

automation
commande refnum, 19-7
interfaces personnalisées, 19-12

B

barre d'outils, 3-5

barre de menus
masquage, 4-30

barres de défilement
listes déroulantes, 4-17
masquage, 4-30

base de connaissances (KnowledgeBase), D-1

bibliothèque d'instruments
ajout de VIs et commandes, 3-7

bibliothèque PDF, 1-1

bibliothèque utilisateur
ajout de VIs et commandes, 3-7

bibliothèques
ajout de VIs et commandes, 3-7
conversion de répertoires en. *Voir l'Aide LabVIEW.*
conversion en répertoires. *Voir l'Aide LabVIEW.*
enregistrement de VIs sous, 7-13
emplacement suggéré pour, A-3
gestion, 7-14
instrument, A-1

- marquage des VIs comme des VIs de niveau principal. *Voir l'Aide LabVIEW.*
 - organisation des, A-1
 - partagées, 7-16
 - distribution des VIs, 7-16
 - 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, 20-1
 - construction, 7-16
 - distribution des VIs, 7-16
 - binaire
 - création de fichiers, 14-10
 - E/S sur fichiers, 14-4
 - nombres à virgule flottante, 6-12
 - Boîtes de calcul, 21-2
 - illustration, 21-3
 - saisie d'équations, 21-2
 - saisie de déclarations de type C, 21-2
 - variables, 21-3
 - boîtes de dialogue
 - commandes, 4-29
 - utilisation. *Voir l'Aide LabVIEW.*
 - commandes de type menu déroulant, 4-19
 - conception, 4-34
 - étiquettes, 4-29
 - faces-avant distantes, 18-16
 - fichier natif. *Voir l'Aide LabVIEW.*
 - indicateurs, 4-29
 - police, 4-32
 - boîtes de dialogue des fichiers natifs. *Voir l'Aide LabVIEW.*
 - boucles
 - auto-indexation, 8-4
 - construction de tableaux, 8-6
 - contrôle de la synchronisation, 8-12
 - données par défaut, 6-13
 - For, 8-2
 - infinies, 8-4
 - registres à décalage, 8-6
 - utilisation. *Voir l'Aide LabVIEW.*
 - While, 8-3
 - boucles "do". *Voir* boucles While.
 - boucles "repeat-until". *Voir* boucles While.
 - boucles For
 - auto-indexation pour définir le décompte, 8-5
 - contrôle de la synchronisation, 8-12
 - données par défaut, 6-13
 - registres à décalage, 8-6
 - terminaux d'itération, 8-2
 - terminaux de décompte, 8-2
 - utilisation. *Voir l'Aide LabVIEW.*
 - boucles While
 - auto-indexation, 8-6
 - contrôle de la synchronisation, 8-12
 - gestion d'erreur, 6-16
 - 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, 4-4
 - face-avant, 4-14
 - boutons rotatifs
 - Voir aussi* numériques.
 - ajout de rampes de couleurs, 4-14
 - face-avant, 4-12
 - bulles de données
 - affichage pendant l'exécution en mode animation. *Voir l'Aide LabVIEW.*
- ## C
- câblage
 - automatiquement, 5-14
 - guides. *Voir l'Aide LabVIEW.*

- manuel, 5-12, 5-15
- outil, 5-15
- structures. *Voir l'Aide LabVIEW.*
- techniques, 5-33
- unités, 5-27
- câblage automatique, 5-14
- câblage manuel sur le diagramme, 5-15
- cadrons
 - Voir aussi* numériques.
 - ajout de rampes de couleurs, 4-14
 - face-avant, 4-12
- calcul d'équations, 21-1
- cartes instruments
 - configuration, 1-5
- chaîne d'appelants
 - affichage, 6-11
- chaînes, 10-1
 - chaînes déroulantes, 4-15
 - commandes et indicateurs, 4-14
 - type de données (table), 5-3
 - types d'affichage, 10-2
 - comparaison, C-1
 - constantes universelles, 5-6
 - division. *Voir l'Aide LabVIEW.*
 - édition par programmation, 10-3
 - fonctions, 5-9
 - formatage, 10-4
 - spécificateurs, 10-4
 - polymorphisme, B-5
 - remplacement du texte. *Voir l'Aide LabVIEW.*
 - tables, 10-3
 - utilisation de chaînes de formatage. *Voir l'Aide LabVIEW.*
 - valeurs numériques à, 10-5
 - variables globales, 11-6
- chaînes déroulantes, 4-15
- changer les types de 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, 4-16
 - type de données (table), 5-4
 - utilisation. *Voir l'Aide LabVIEW.*
 - constantes universelles, 5-6
 - E/S sur fichiers, 14-7
 - faces-avant distantes, 18-17
 - invalides, 4-16
 - options, 3-9
 - vides, 4-16
- chemins de répertoires. *Reportez-vous à chemins.*
- chemins de répertoires. *Voir sondes.*
- chemins invalides, 4-16
- chemins vides, 4-16
- chevauchement d'objets de la face-avant, 4-22
- CIN, 20-2
- client
 - formation, D-1
 - services professionnels, D-1
 - support technique, D-1
- clients
 - ActiveX, 19-8
 - LabVIEW pour les faces-avant distantes, 18-15
 - multiples pour les faces-avant distantes, 18-14
 - Navigateur Web pour des faces-avant distantes, 18-15
 - .NET, 19-4
- clonage des objets de la face-avant ou du diagramme. *Voir l'Aide LabVIEW.*
- clusters
 - commandes et indicateurs, 4-17
 - type de données (table), 5-4
 - comparaison, C-2
 - conversion des tableaux. *Voir l'Aide LabVIEW.*
 - déplacement. *Voir l'Aide LabVIEW.*

- erreur, 6-15
 - composants, 6-16
 - rapports. *Voir l'Aide LabVIEW.*
- fonctions, 5-10
- modèles de fils de liaison, 10-16
- ordre des éléments
 - modification, 10-16

Voir aussi l'Aide LabVIEW.
- polymorphisme, B-6
- redimensionnement. *Voir l'Aide LabVIEW.*
- code C
 - appel à partir de LabVIEW, 20-2
- Code Interface Node, 20-2
- code source. *Voir* diagramme.
- codes d'erreur définis par l'utilisateur. *Voir l'Aide LabVIEW.*
- coller
 - graphiques, 4-6
- coloriage
 - couleurs système, 4-34

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, 4-10
 - 3D, 4-10
 - ActiveX, 19-7
 - affichage des éléments optionnels, 4-2
 - ajouter à des bibliothèques, 3-7
 - appellation, 7-12
 - boîte de couleur, 4-13
 - Booléen, 4-14
 - utilisation. *Voir l'Aide LabVIEW.*
- chaîne, 4-14
 - tables, 10-3
 - types d'affichage, 10-2
- changement en indicateurs, 4-2
- chemin, 4-16
 - utilisation. *Voir l'Aide LabVIEW.*
- classiques, 4-10
- cluster, 4-17
- coloriage, 4-5
- conception d'interfaces utilisateur, 4-33
- création dans un diagramme. *Voir l'Aide LabVIEW.*
- définitions de type, 4-2
- dialogue, 4-29
 - utilisation. *Voir l'Aide LabVIEW.*
- directives pour l'utilisation de la face-avant, 4-33
- glissières, 4-11
- groupement et verrouillage, 4-7
- horodatage, 4-13
- icônes, 5-2
- impression, 15-3
- liste déroulante, 4-17
 - utilisation. *Voir l'Aide LabVIEW.*
- masquage
 - Voir aussi l'Aide LabVIEW.*
 - éléments optionnels, 4-2
- masqués. *Voir l'Aide LabVIEW.*
- menu déroulant, 4-19
 - utilisation. *Voir l'Aide LabVIEW.*
- nécessaires, 7-8
- nom d'E/S, 4-23
- nombre de couleurs élevé, 4-10
- nombre de couleurs réduit, 4-10
- numérique, 4-11
 - utilisation. *Voir l'Aide LabVIEW.*
- numériques, 4-12
- onglet, 4-22
- optionnels, 7-8
- palette, 3-1

- navigation et recherche, 3-2
- personnalisation, 3-6
- raccourcis clavier, 4-4
- rampe de couleurs, 4-13
- redimensionnement, 4-7
 - par rapport à la taille de la fenêtre, 4-8
- refnum, 4-29
 - utilisation. *Voir l'Aide LabVIEW.*
- refnum automation, 19-7
- remplacement, 4-3
- rotatifs, 4-12
- sous-titres pour les info-bulles des sous-VIs. *Voir l'Aide LabVIEW.*
- style, 16-2
- sur le diagramme, 5-1
- tableau, 4-17
- terminaux
 - icônes, 5-2
 - types de données, 5-2
- terminaux (table), 5-3
- terminaux de type de données, 5-2
- type énumération, 4-20
 - avancées, 4-21
 - utilisation. *Voir l'Aide LabVIEW.*
- types de données (table), 5-3
- commandes arbre, 4-18
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes de face-avant secondaire, 4-22
- commandes de type
 - énumération, 4-20
 - avancées, 4-21
- commandes de type énumération, 4-20
 - avancées, 4-21
 - type de données (table), 5-3
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes de type liste déroulante, 4-17
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes de type menu déroulant, 4-19
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes de type menu déroulant d'images, 4-19
- commandes et indicateurs 2D, 4-10
- commandes et indicateurs 3D, 4-10
- commandes et indicateurs à glissières, 4-11
 - Voir aussi* numériques.
- commandes et indicateurs booléens, 4-14
 - comparaison de valeurs, C-1
 - type de données (table), 5-3
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes et indicateurs classiques, 4-10
- commandes et indicateurs de type image
 - type de données (table), 5-5
 - utilisation, 13-1
- commandes et indicateurs numériques, 4-12
- commandes et indicateurs rotatifs, 4-12
- commandes onglet, 4-22
 - utilisation. *Voir l'Aide LabVIEW.*
- commandes système, 18-22
- commentaire excluant une section particulière du diagramme
 - mise au point des VIs, 6-11
- communication, 18-1
 - ActiveX, 19-1
 - Apple events, 18-22
 - bas niveau, 18-21
 - canaux, 18-22
 - DataSocket, 18-2
 - E/S sur fichiers, 14-1
 - exécution de commandes au niveau système, 18-22
 - fonctions, 7-4
 - Mac OS, 18-22
 - PPC, 18-22
 - protocoles, 18-21
 - TCP, 18-21
 - UDP, 18-21
 - UNIX, 18-22
 - VI Serveur, 17-1
 - VI System Exec, 18-22
 - VIs, 7-4
- communication de bas niveau, 18-21

- communication de programme à programme, 18-22
- communication par canaux, 18-22
- commutateurs sur la face-avant, 4-14
- 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, 4-14
 - face-avant, 4-12
- conception
 - boîtes de dialogue, 4-34
 - diagramme, 5-32
 - face-avant, 4-33
 - projets, 7-1
 - sous-VIs, 7-11
- conception du projet, 7-1
- conditions normales comme erreurs. *Voir l'Aide LabVIEW.*
- conditions par défaut, 8-13
- configuration
 - apparence et comportement des VIs, 16-1
 - des commandes de la face-avant, 4-1
 - des indicateurs de la face-avant, 4-1
 - événements dynamiques. *Voir l'Aide LabVIEW.*
 - événements utilisateur. *Voir l'Aide LabVIEW.*
 - face-avant, 4-3
 - menus, 16-2
 - serveurs pour les faces-avant
 - distantes, 18-13
 - LabVIEW, 18-15
 - navigateur Web, 18-16
- connecteurs, 2-5
 - définition, 7-7
 - entrées et sorties nécessaires et optionnelles, 7-8
 - impression, 15-3
- connexion des terminaux, 5-12
- constantes, 5-5
 - création. *Voir l'Aide LabVIEW.*
 - définies par l'utilisateur, 5-6
 - définition de paramètres avec ActiveX, 19-12
 - édition. *Voir l'Aide LabVIEW.*
 - tableaux, 10-12
 - universelles, 5-6
- constantes définies par l'utilisateur, 5-6
- constantes universelles, 5-6
- construction
 - applications autonomes
 - distribution des VIs, 7-16
 - applications des drivers d'instrument. *Voir l'Aide LabVIEW.*
 - applications du VI Serveur, 17-2
 - bibliothèques partagées
 - distribution des VIs, 7-16
 - diagramme, 5-1
 - face-avant, 4-1
 - sous-VIs, 7-5
 - VIs, 7-1
 - VIs polymorphes, 5-19
- contacter National Instruments, D-2
- Conteneur ActiveX
 - mode de conception, 19-9
- conteneurs, 4-21
 - ActiveX, 19-7
 - commandes de face-avant
 - secondaire, 4-22
 - commandes onglet, 4-22
- contrôle
 - code source, 7-2
 - instruments, 7-4

- objets de la face-avant à distance. *Voir l'Aide LabVIEW.*
- objets de la face-avant par programmation, 17-9
- par programmation des VIs, 17-1
- VIs à distance, 18-13
- VIs lorsqu'ils sont appelés comme sous-VIs, 7-4
- contrôle de code source, 7-2
- conventions utilisées dans ce manuel, xxii
- 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, 10-7
 - valeurs numériques en chaînes, 10-5
 - XML en données LabVIEW, 10-8
- conversion de VIs Express en sous-VIs, 5-22
Voir aussi l'Aide LabVIEW.
- copie
 - graphiques, 4-6
 - objets sur la face-avant ou le diagramme.
Voir l'Aide LabVIEW.
 - VIs, A-3
- correction
 - fils de liaison brisés, 5-16
 - VIs, 6-2
 - techniques de mise au point, 6-4
- couleur
 - boîtes, 4-13
 - commandes et indicateurs utilisant un nombre de couleurs élevé, 4-10
 - commandes et indicateurs utilisant un nombre réduit de couleurs, 4-10
 - création dans des graphiques, 13-6
 - mappage, 12-15
 - modification dans des graphiques, 13-6
 - options, 3-9
 - rampes
 - commandes et indicateurs rotatifs, 4-14
 - sélecteur, 4-13
 - 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 *Voir couleurs système.*
 - couleurs système, 4-34
Voir aussi l'Aide LabVIEW.
 - courbes
 - ajout à des graphes et graphes déroulants.
Voir l'Aide LabVIEW.
 - empilées, 12-8
 - lissage, 12-2
 - superposées, 12-8
 - courbes empilées, 12-8
 - courbes lissées, 12-2
 - courbes superposées, 12-8
 - création
 - constantes définies par l'utilisateur, 5-6
 - descriptions des objets, 15-3
 - descriptions des VIs, 15-3
 - événements utilisateur, 9-15
 - fichiers binaires, 14-10
 - fichiers journaux, 14-11
 - fichiers tableur, 14-9
 - fichiers texte, 14-9
 - graphes déroulants. *Voir l'Aide LabVIEW.*
 - graphes. *Voir l'Aide LabVIEW.*
 - historique des révisions, 15-2
 - icônes, 7-9
 - info-bulles, 15-3
 - menus, 16-3
 - références des commandes. *Voir l'Aide LabVIEW.*
 - sous-palettes. *Voir l'Aide LabVIEW.*

sous-VIs, 7-11
 situations à éviter. *Voir l'Aide LabVIEW.*
 tableaux, 10-12
 types de palettes, 3-7
 création de sous-VIs à partir de VIs
 Express, 5-22
Voir aussi l'Aide LabVIEW.
 créer
 Objet .NET *Voir l'Aide LabVIEW.*
 curseurs
 ajout aux graphes. *Voir l'Aide LabVIEW.*
 graphe, 12-4
 suppression dans les graphes. *Voir l'Aide LabVIEW.*

D

d'événements dynamiques
 exemple, 9-13
 DAQ
 Afficheur de voies virtuelles, 1-5
 Assistant Entrées/Sorties, 1-5
 Assistant Solutions, 1-5
 transmission de noms de voie, 4-23
 Utilitaire de configuration, 1-5
 VIs et fonctions, 7-4
 DataSocket, 18-2
 contrôle des objets de la face-avant. *Voir l'Aide LabVIEW.*
 diagramme, 18-7
 données bufférisées, 18-8
 données variant, 18-10
 face-avant, 18-5
 fermeture de connexions par programmation, 18-8
 ouverture de connexions par programmation, 18-8
 pour les données, 18-5
 protocoles, 18-3
 URL, 18-3

déclarations. *Voir* nœuds.
 défilement
 graphes, 12-3
 graphes déroulants, 12-3
 défilement dans un diagramme, 12-3
 définition
 codes d'erreur. *Voir l'Aide LabVIEW.*
 des couleurs utilisateur. *Voir l'Aide LabVIEW.*
 définitions de type, 4-2
 regroupement des objets de la face-avant, 4-7
 dépannage. *Voir* mise au point.
 dépassement des limites inférieures pour les nombres, 6-12
 dépassement des limites supérieures, 6-12
 dépendance artificielle des données, 5-31
 dépendance des données, 5-30
 absente, 5-31
 artificielle, 5-31
 contrôle avec des structures
 Séquence, 8-17
 paramètres dupliqués, 14-13
 situations de compétition, 11-5
 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.*
 désactivation
 momentanée du routage
 automatique, 5-16
 outils de mise au point, 6-12
 sections d'un diagramme
 mise au point des VIs, 6-11
 désenregistrement
 dynamique des événements, 9-12
 événements utilisateur, 9-17
 développement de VIs, 7-1
 directives, 1-3

- suivi du développement. *Reportez-vous à documentation des VIs.*
- déverrouillage
 - fenêtre Aide contextuelle, 3-5
 - objets de la face-avant, 4-7
 - VIs. *Voir l'Aide LabVIEW.*
- diagramme, 2-2
 - ajout d'espace sans redimensionner, 5-33
 - alignement des objets, 4-6
 - Voir aussi l'Aide LabVIEW.*
 - cablage automatique, 5-14
 - câblage manuel, 5-12, 5-15
 - commentaire excluant une section particulière, 6-11
 - conception, 5-32
 - constantes, 5-5
 - contrôler le code source, 7-2
 - copie d'objets. *Voir l'Aide LabVIEW.*
 - création de commandes et d'indicateurs. *Voir l'Aide LabVIEW.*
 - DataSocket, 18-7
 - données variant, 5-25
 - espacement d'objets de façon régulière, 4-6
 - Voir aussi l'Aide LabVIEW.*
 - étiquettes, 4-30
 - création. *Voir l'Aide LabVIEW.*
 - édition. *Voir l'Aide LabVIEW.*
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - flux de données, 5-29
 - fonctions, 5-8
 - impression, 15-6
 - insertion d'objets. *Voir l'Aide LabVIEW.*
 - nœuds, 5-7
 - objets, 5-1
 - options, 3-9
 - planification, 7-1
 - points de coercition, 5-17
 - polices, 4-31
 - protection par mot de passe, 7-16
 - recherche de terminaux. *Voir l'Aide LabVIEW.*
 - remplacement des objets. *Voir l'Aide LabVIEW.*
 - réorganisation d'objets. *Voir l'Aide LabVIEW.*
 - répartition des objets, 4-6
 - Voir aussi l'Aide LabVIEW.*
 - structures, 8-1
 - utilisation. *Voir l'Aide LabVIEW.*
 - suppression d'objets. *Voir l'Aide LabVIEW.*
 - terminaux
 - affichage, 5-2
 - ajout de fonctions, 5-12
 - commandes et indicateurs (table), 5-3
 - objets de la face-avant et, 5-1
 - suppressions en provenance de fonctions, 5-12
 - types de données (table), 5-3
 - VI Serveur, 17-1
 - dimensionnement. *Voir redimensionnement.*
 - dimensions
 - tableaux, 10-9
 - directives de développement, 1-3
 - division
 - chaînes. *Voir l'Aide LabVIEW.*
 - DLLs
 - appel à partir de LabVIEW, 20-1
 - construction
 - distribution des VIs, 7-16
 - documentation
 - bibliothèque en ligne, D-1
 - bibliothèque PDF, 1-1
 - conventions utilisées dans ce manuel, xxii
 - guide, 1-1
 - introduction à ce manuel, xxi
 - organisation de ce manuel, xxii
 - structure des répertoires, A-2
 - utilisation avec d'autres ressources, 1-1
 - utilisation de ce manuel, xxi

- documentation des VIs
 - création d'info-bulles, 15-3
 - création de descriptions des VIs et objets, 15-3
 - fichiers d'aide, 15-5
 - historique des révisions, 15-2
 - impression, 15-3
 - lien vers des fichiers d'aide que vous créez. *Voir l'Aide LabVIEW.*
 - par programmation, 15-4
- données
 - envoi par e-mail de VIs, 18-18
- données aplaties
 - données variant et, 5-26
- données bufférisées
 - DataSocket, 18-8
 - variables locales, 11-6
- données inattendues *Voir données par défaut;*
données indéfinies.
- données indéfinies
 - détection, 6-12
 - éviter, 6-14
 - Inf (infinity), 6-12
 - NaN (not a number), 6-12
 - tableaux, 6-13
- données numériques
 - acquisition d'un sous-ensemble, 4-26
 - ajout, 4-28
 - compression, 4-28
 - recherche d'un pattern, 4-28
 - type de données waveform numérique, 12-21
- données par défaut
 - boucles For, 6-13
 - répertoire, 14-22
 - tableaux, 6-13
- données variant
 - ActiveX, 19-7
 - attributs, 5-26
 - commandes et indicateurs
 - type de données (table), 5-4

- conversion en, 5-26
- DataSocket, 18-10
- données aplaties et, 5-26
- édition des attributs. *Voir l'Aide LabVIEW.*
- gestion, 5-25
- données waveform
 - commandes et indicateurs
 - type de données (table), 5-4
 - fonctions, 5-11
 - graphes, 12-9
 - graphiques, 13-3
 - types de données, 12-10
 - graphes déroulants, 12-12
 - type de données, 12-20
- drivers
 - instrument, D-1
 - LabVIEW. *Voir l'Aide LabVIEW.*
 - logiciel, D-1
- drivers d'instrument
 - LabVIEW. *Voir l'Aide LabVIEW.*
- drivers d'instruments, D-1
- drivers logiciels, D-1

E

- E/S
 - commandes et indicateurs, 4-23
 - type de données (table), 5-5
 - commandes et indicateurs de nom, 4-23
 - erreur, 6-16
 - fichier. *Voir E/S sur fichiers.*
- E/S sur fichier journal, 14-4
 - création de fichiers, 14-11
- E/S sur fichiers
 - chemins, 14-7
 - écriture de waveforms, 14-11
 - enregistrement de données de la face-avant, 14-16
 - enregistrement sur disque en continu, 14-8

- fichier .lvm, 14-22
- fichiers binaires, 14-4
 - création, 14-10
- fichiers journaux, 14-4
 - création, 14-11
- fichiers standard, 14-4
- fichiers tableur
 - création, 14-9
- fichiers texte, 14-2
 - création, 14-9
- fonctions, 5-10
- fonctions de fichiers avancées, 14-7
- format des données LabVIEW, 14-22
- formats, 14-2
- lecture de waveforms, 14-12
- mise en réseau et, 18-1
- opération de base, 14-1
- paramètres dupliqués, 14-13
- refnums, 14-1
- répertoire de données par défaut, 14-22
- sélection du répertoire par défaut. *Voir l'Aide LabVIEW.*
- VIs de fichiers de configuration
 - format, 14-15
 - lecture et écriture de fichiers
 - .ini, 14-13
 - spécial, 14-14
- VIs de haut niveau, 14-6
- VIs et fonctions de bas niveau, 14-7
- échelles x
 - multiples, 12-2
- échelles y
 - multiples, 12-2
- écriture dans des fichiers, 14-1
- Éditeur de menu, 16-3
- édition
 - étiquettes. *Voir l'Aide LabVIEW.*
 - menus, 16-2
 - menus locaux des VIs polymorphes. *Voir l'Aide LabVIEW.*
 - types de palettes, 3-7
- effacement
 - graphes et graphes déroulants. *Voir l'Aide LabVIEW.*
 - indicateurs. *Voir l'Aide LabVIEW.*
- éléments de menus MRU, 3-4
- éléments de menus utilisés le plus récemment, 3-4
- éliminer des enregistrements de données, 14-18
- e-mail
 - envoi à partir de VIs, 18-18
 - jeux de caractères, 18-18
 - translitération, 18-20
- emplacement pour enregistrer des fichiers, A-3
- enregistrement
 - dynamique d'événements, 9-10
 - dynamique d'événements. *Voir l'Aide LabVIEW.*
 - événements utilisateur, 9-15
 - événements utilisateur. *Voir l'Aide LabVIEW.*
 - options de l'environnement de travail, 3-9
 - statique des événements, 9-9
- enregistrement dans un fichier journal
 - automatique, 14-17
 - changer la liaison vers le fichier journal, 14-19
 - effacer la liaison vers le fichier journal, 14-18
 - extraction des données par programmation, 14-19
 - interactif, 14-17
 - suppression d'enregistrements, 14-18
- enregistrement de données dans un fichier journal. *Voir enregistrement dans un fichier journal.*
- enregistrement de fichiers
 - emplacement suggéré pour, A-3
- enregistrement de la configuration des VIs Express en tant que VIs, 5-22

- Voir aussi l'Aide LabVIEW.*
- enregistrement des VIs
 - au format antérieur, 7-15
 - bibliothèques, 7-13
 - fichiers individuels, 7-13
 - revenir aux dernières versions enregistrées. *Voir l'Aide LabVIEW.*
- enregistrement sur disque en continu, 14-8
- enregistrements, 14-16
 - spécification pendant la récupération des données de la face-avant au moyen des sous-VIs, 14-20
 - suppression, 14-18
- entiers
 - conversion, B-1
 - dépassement des limites inférieures et supérieures, 6-12
- équations
 - Boîtes de calcul, 21-2
 - intégration dans LabVIEW, 21-1
 - MATLAB
 - nœud de script, 21-5
 - scripts de mise au point, 21-6
 - méthodes d'utilisation, 21-1
 - nœuds d'expression, 21-4
- erreurs
 - acheminement particulier. *Voir l'Aide LabVIEW.*
 - affichage, 6-2
 - annulation d'erreurs existantes. *Voir l'Aide LabVIEW.*
 - clusters, 6-15
 - composants, 6-16
 - connecteur, 7-8
 - rapports. *Voir l'Aide LabVIEW.*
 - codes, 6-15
 - conditions normales comme. *Voir l'Aide LabVIEW.*
 - définition d'erreurs personnalisées. *Voir l'Aide LabVIEW.*
 - détection, 6-15
 - E/S, 6-16
 - fenêtre, 6-2
 - gestion, 6-14
 - commande d'instrument. *Voir l'Aide LabVIEW.*
 - méthodes, 6-15
 - utilisation des boucles While, 6-16
 - utilisation des structures
 - Condition, 6-17
 - gestion automatique, 6-14
 - liste, 6-2
 - notification. *Voir l'Aide LabVIEW.*
 - recherche, 6-2
 - techniques de mise au point, 6-4
 - unités incompatibles, 5-27
 - VIs brisés, 6-2
- espace
 - ajout à la face-avant ou au diagramme, 4-10
- espace disque
 - options, 3-9
 - vérification. *Voir l'Aide LabVIEW.*
- espacement d'objets de façon régulière, 4-6
 - Voir aussi l'Aide LabVIEW.*
- étiquetage
 - constantes, 5-5
 - création d'étiquettes libres. *Voir l'Aide LabVIEW.*
 - édition. *Voir l'Aide LabVIEW.*
 - polices, 4-31
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - sous-titres, 4-31
 - unités de mesures, 5-27
 - variables globales, 11-3
 - variables locales, 11-2
- étiquettes
 - affichage des constantes automatiques. *Voir l'Aide LabVIEW.*
 - boîte de dialogue, 4-29
 - transparents. *Voir l'Aide LabVIEW.*

- étiquettes d'unité, 5-27
- étiquettes dépendantes, 4-30
 - édition. *Voir l'Aide LabVIEW.*
- étiquettes des constantes automatiques
 - affichage. *Voir l'Aide LabVIEW.*
- étiquettes libres, 4-30
 - création. *Voir l'Aide LabVIEW.*
- étirement. *Voir redimensionnement.*
- événement de type filtre, 9-5
- événements
 - Voir aussi* structures événement.
 - ActiveX, 19-7, 19-14
 - configuration. *Voir l'Aide LabVIEW.*
 - création d'événements, 9-15
 - définition, 9-1
 - désenregistrement dynamique, 9-12
 - désenregistrement utilisateur, 9-17
 - disponibles dans LabVIEW. *Voir l'Aide LabVIEW.*
 - dynamique
 - enregistrement, 9-10
 - dynamiques
 - exemple, 9-13
 - enregistrement dynamique. *Voir l'Aide LabVIEW.*
 - enregistrement statique, 9-9
 - filtre, 9-5
 - génération d'événements, 9-16
 - gestion, 9-6
 - gestion ActiveX, 19-14
 - notification, 9-5
 - pris en charge, 9-1
 - utilisateur, 9-15
 - verrouillage de la face-avant, 9-8
- événements de type notification, 9-5
- événements dynamiques
 - enregistrement, 9-10
 - enregistrement *Voir l'Aide LabVIEW.*
- événements utilisateur
 - Voir aussi l'Aide LabVIEW.*
 - création, 9-15
 - désenregistrement, 9-17
 - enregistrement. *Voir l'Aide LabVIEW.*
 - exemple, 9-17
 - génération, 9-16
- exécution
 - animation
 - afficher les bulles de données. *Voir l'Aide LabVIEW.*
 - mise au point des VIs, 6-6
 - sondage automatique. *Voir l'Aide LabVIEW.*
 - flux, 5-29
 - contrôle avec des structures Séquence, 8-17
 - suspension
 - mise au point des VIs, 6-10
- exécution de commandes au niveau système, 18-22
- exécution des VIs, 6-1
- exécution des VIs par incrémentation, 6-6
- exécution en continu des VIs, 6-1
- exemple de code, D-1
- exemples, 1-5
 - tableaux, 10-9
 - tableaux 1D, 10-9
 - tableaux 2D, 10-10

F

- face-avant, 2-1
 - affichage avec différentes résolutions d'écran, 4-35
 - affichage des éléments optionnels des objets, 4-2
 - ajout d'espace sans redimensionner, 4-10
 - alignement des objets, 4-6
 - Voir aussi l'Aide LabVIEW.*
 - caractéristiques du texte, 4-31
 - changement de commandes en indicateurs et vice versa, 4-2

- chargement par le biais des commandes de face-avant secondaire, 4-22
- chevauchement d'objets, 4-22
- coloriage des objets, 4-5
 - arrière-plan et premier plan. *Voir l'Aide LabVIEW.*
 - copie des couleurs. *Voir l'Aide LabVIEW.*
- commandes, 4-10
- conception, 4-33
- contrôle à distance, 18-13
- contrôle d'objets à distance. *Voir l'Aide LabVIEW.*
- contrôler les objets par programmation, 17-9
- copie d'objets. *Voir l'Aide LabVIEW.*
- DataSocket, 18-5
- définition de l'ordre de défilement par tabulation, 4-5
- définition de la taille de la fenêtre. *Voir l'Aide LabVIEW.*
- définitions de type, 4-2
- effacement des indicateurs. *Voir l'Aide LabVIEW.*
- enregistrement dans un fichier journal, 14-16
- enregistrement de données, 14-16
- espacement d'objets de façon régulière, 4-6
 - Voir aussi l'Aide LabVIEW.*
- étiquettes, 4-30
 - création. *Voir l'Aide LabVIEW.*
 - édition. *Voir l'Aide LabVIEW.*
 - redimensionnement. *Voir l'Aide LabVIEW.*
- groupement et verrouillage des objets, 4-7
- importation de graphiques, 4-6
- impression, 15-6
 - après exécution du VI, 15-7
- indicateurs, 4-10
- insertion d'objets à l'aide d'ActiveX, 19-9
- masquage
 - éléments optionnels des objets, 4-2
 - objets. *Voir l'Aide LabVIEW.*
- mise à l'échelle des objets, 4-8
- objets
 - terminaux du diagramme et, 5-1
 - objets masqués. *Voir l'Aide LabVIEW.*
 - objets transparents. *Voir l'Aide LabVIEW.*
- options, 3-9
- ordre des objets, 4-5
- planification, 7-1
- polices, 4-31
- publication d'images sur le Web, 18-12
- raccourcis clavier, 4-4
- recherche d'objets. *Voir l'Aide LabVIEW.*
- récupération des données
 - utilisation de sous-VIs, 14-19
 - utilisation des fonctions d'E/S sur fichier, 14-21
- redimensionnement des objets, 4-7
 - par rapport à la taille de la fenêtre, 4-8
- remplacement des objets, 4-3
- réorganisation d'objets. *Voir l'Aide LabVIEW.*
- répartition des objets, 4-6
 - Voir aussi l'Aide LabVIEW.*
- sous-titres, 4-31
 - création. *Voir l'Aide LabVIEW.*
- sous-VIs, 7-11
- style des commandes et des indicateurs, 16-2
- suppression d'objets. *Voir l'Aide LabVIEW.*
- verrouillage avec des événements, 9-8
- visualisation à distance, 18-13
- faire glisser et déposer. *Voir l'Aide LabVIEW.*
- fenêtre Aide contextuelle, 3-5
 - apparence des terminaux, 7-9
 - création de descriptions d'objets, 15-3
 - création de descriptions de VIs, 15-3

- Fenêtre de hiérarchie, 7-12
 - impression, 15-3
 - recherche. *Voir l'Aide LabVIEW.*
- fichier .lvm, 14-22
- fichier de menu d'exécution, 16-3
- fichier rtm, 16-3
- fichiers bitmap, 13-7
- fichiers BMP, 13-7
- fichiers d'aide, 1-2
 - création de vos propres, 15-5
 - HTML, 15-5
 - lien vers des VIs. *Voir l'Aide LabVIEW.*
 - RTF, 15-5
- fichiers GIF, 15-5
- fichiers .ini
 - lecture et écriture, 14-13
- fichiers Joint Photographic Experts Group, 13-7
- fichiers JPEG, 13-7, 15-4
 - Serveur Web, 18-13
- fichiers PNG, 13-7, 15-4
 - Serveur Web, 18-13
- fichiers Portable Network Graphics, 13-7
- fichiers standard, 14-4
- fichiers tableur
 - création, 14-9
 - écriture de données numériques sur des, 10-5
- fichiers texte
 - création, 14-9
 - E/S sur fichiers, 14-2
 - écriture de données numériques sur des, 10-5
 - enregistrement de la documentation dans des, 15-4
- files d'attente
 - données variant, 5-26
- fils de liaison, 2-4
 - brisés, 5-16
 - déplacement. *Voir l'Aide LabVIEW.*
 - routage, 5-15
 - sélection, 5-16
- fils de liaison brisés, 5-16
- flux d'exécution, 5-29
- flux de données
 - observation, 6-6
- fonctions, 5-8
 - ajouts de terminaux, 5-12
 - avancées, 5-11
 - Booléen, 5-9
 - chaîne, 5-9
 - cluster, 5-10
 - contrôle d'applications, 5-11
 - diagramme, 5-8
 - dialogue, 5-10
 - données waveform, 5-11
 - E/S sur fichiers, 5-10
 - numérique, 5-8
 - palette
 - navigation et recherche, 3-2
 - personnalisation, 3-6
 - 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, 5-12
 - tableau, 5-9
 - temps, 5-10
- fonctions avancées, 5-11
- Fonctions booléennes, 5-9
 - polymorphisme, B-4
- Fonctions de comparaison, C-1
 - polymorphisme, B-6
- fonctions de contrôle d'applications, 5-11
- fonctions dialogue, 5-10
- Fonctions logarithmiques
 - polymorphisme, B-8
- fonctions temps, 5-10

formatage
 chaînes, 10-4
 spécificateurs, 10-4
 texte sur la face-avant, 4-31
formatage des caractères, 4-31
formater des chaînes de caractères *Voir l'Aide LabVIEW.*
formation
 client, D-1
formats pour l'E/S sur fichier
 fichiers binaires, 14-4
 fichiers journaux, 14-4
 fichiers texte, 14-2
formules. *Voir équations.*

G

génération
 événements utilisateur, 9-16
génération de rapports, 15-8
 clusters d'erreur. *Voir l'Aide LabVIEW.*
gestion des événements, 9-6
glissières
 ajout, 4-11
GPIB
 configuration, 1-5
 graphe à balayage, 12-7
 graphe déroulant, 12-7
 graphe numérique
 configuration de courbes. *Voir l'Aide LabVIEW.*
 affichage de données numériques, 12-16
graphes, 12-1
 3D, 12-19
 abaques de Smith, 13-3
 ajout de tracés. *Voir l'Aide LabVIEW.*
 courbes lissées, 12-2
 création. *Voir l'Aide LabVIEW.*
 curseurs, 12-4
 ajout. *Voir l'Aide LabVIEW.*
 suppression. *Voir l'Aide LabVIEW.*

 d'intensité, 12-13
 options, 12-16
 défilement, 12-3
 données waveform, 12-9
 types de données, 12-10
 échelles multiples, 12-2
 effacement. *Voir l'Aide LabVIEW.*
 formatage de l'échelle, 12-5
 graphiques, 13-2
 lignes de transmission, 13-3
 mise à l'échelle, 12-5
 numériques
 masquage des données, 12-19
 options, 12-2
 personnalisation de l'apparence, 12-3
 personnalisation du comportement, 12-3
 polaires, 13-3
 rafraîchissements progressifs, 12-6
 types, 12-1
 XY, 12-9
 types de données, 12-11, 12-12
 zoom. *Voir l'Aide LabVIEW.*
graphes 3D, 12-19
graphes d'intensité, 12-13
 options, 12-16
 représentation des couleurs, 12-15
graphes déroulants, 12-1
 ajout de tracés. *Voir l'Aide LabVIEW.*
 courbes empilées, 12-8
 courbes lissées, 12-2
 courbes superposées, 12-8
 création. *Voir l'Aide LabVIEW.*
 d'intensité, 12-13
 options, 12-16
 défilement, 12-3
 données waveform, 12-12
 échelles multiples, 12-2
 effacement. *Voir l'Aide LabVIEW.*
 formatage de l'échelle, 12-5
 longueur de l'historique, 12-7

- options, 12-2
- personnalisation de l'apparence, 12-3
- personnalisation du comportement, 12-7
- types, 12-1
- zoom. *Voir l'Aide LabVIEW.*
- graphes déroulants d'intensité, 12-13
 - options, 12-16
 - représentation des couleurs, 12-15
- graphes numériques
 - courbes lissées, 12-2
 - masquage des données, 12-19
- graphes polaires, 13-3
- graphes XY, 12-9
 - types de données, 12-11, 12-12
- graphiques
 - ajout à l'icône du VI. *Voir l'Aide LabVIEW.*
 - commandes et indicateurs de type image
 - type de données (table), 5-5
 - utilisation, 13-1
 - création de couleurs, 13-6
 - entrée de texte, 13-4
 - faire glisser et déposer. *Voir l'Aide LabVIEW.*
 - formats, 13-7
 - pour fichiers HTML, 15-4
 - graphes, 13-2
 - importation, 4-6
 - modification des couleurs, 13-6
 - publication de faces-avant sur le Web, 18-12
 - tables de pixels, 13-4
 - tracé de formes, 13-4
- grille, 4-6
 - options, 3-9
- groupement
 - données
 - chaînes, 10-1
 - clusters, 10-15
 - tableaux, 10-9

- objets de la face-avant, 4-7
- VI dans des bibliothèques, 7-13

H

- historique
 - Reportez-vous aussi à historique des révisions.*
 - graphes déroulants, 12-7
 - options, 3-9
- historique des révisions
 - création, 15-2
 - impression, 15-3
 - nombre, 15-2
- horodatage
 - Voir aussi numériques.*
 - commandes et indicateurs, 4-13
 - types de données (table), 5-3
- HTML
 - Voir aussi Web.*
 - création de documents, 18-12
 - enregistrement de la documentation dans des, 15-4
 - fichiers d'aide, 15-5
 - formats graphiques, 15-4
 - génération de rapports, 15-8
 - publication de VIs sur le Web, 18-11

I

- icônes, 2-5
 - création, 7-9
 - édition, 7-9
 - impression, 15-3
 - sous-VIs, 7-10
 - VIs Express, 7-10
- ignorer les touches de fonction par défaut. *Voir l'Aide LabVIEW.*
- images animées des faces-avant, 18-12
- images statiques de faces-avant, 18-12
- images. *Reportez-vous à graphiques.*

- impédance des lignes de transmission, 13-4
- importation de graphiques, 4-6
- impression
 - de données à partir d'un VI de plus haut niveau, 15-8
 - de la face-avant après exécution du VI, 15-7
 - documentation des VIs, 15-3
 - enregistrement de la documentation
 - dans des fichiers texte, 15-4
 - dans un fichier HTML, 15-4
 - dans un fichier RTF, 15-4
 - fenêtre active, 15-6
 - options, 3-9
 - par programmation, 15-7
 - rapports, 15-8
 - techniques, 15-9
 - utilisation de sous-VIs, 15-8
- indexation des boucles, 8-4
 - boucles For, 8-5
 - boucles While, 8-6
- indicateurs, 4-1
 - 2D, 4-10
 - 3D, 4-10
 - ActiveX, 19-7
 - affichage des éléments optionnels, 4-2
 - boîte de couleur, 4-13
 - Booléen, 4-14
 - utilisation. *Voir l'Aide LabVIEW.*
 - chaîne, 4-14
 - types d'affichage, 10-2
 - changement en commandes, 4-2
 - chemin, 4-16
 - utilisation. *Voir l'Aide LabVIEW.*
 - classiques, 4-10
 - cluster, 4-17
 - coloriage, 4-5
 - conception d'interfaces utilisateur, 4-33
 - création dans un diagramme. *Voir l'Aide LabVIEW.*
 - définitions de type, 4-2
 - dialogue, 4-29
 - directives pour l'utilisation de la face-avant, 4-33
 - effacement. *Voir l'Aide LabVIEW.*
 - glissières, 4-11
 - groupement et verrouillage, 4-7
 - horodatage, 4-13
 - icônes, 5-2
 - impression, 15-3
 - masquage
 - Voir aussi l'Aide LabVIEW.*
 - éléments optionnels, 4-2
 - masqués. *Voir l'Aide LabVIEW.*
 - nécessaires, 7-8
 - nom d'E/S, 4-23
 - nombre de couleurs élevé, 4-10
 - nombre de couleurs réduit, 4-10
 - numérique, 4-11
 - utilisation. *Voir l'Aide LabVIEW.*
 - numériques, 4-12
 - onglet, 4-22
 - optionnels, 7-8
 - rampe de couleurs, 4-13
 - redimensionnement, 4-7
 - par rapport à la taille de la fenêtre, 4-8
 - refnum, 4-29
 - utilisation. *Voir l'Aide LabVIEW.*
 - remplacement, 4-3
 - rotatifs, 4-12
 - style, 16-2
 - sur le diagramme, 5-1
 - tableau, 4-17
 - terminaux
 - icônes, 5-2
 - types de données, 5-2
 - terminaux (table), 5-3
 - terminaux de type de données, 5-2
 - type énumération
 - avancées, 4-21
 - types de données (table), 5-3

- indices des tableaux, 10-9
 - affichage, 10-13
- Inf (infinity) valeur flottante
 - données indéfinies, 6-12
- info-bulles
 - affichage sur les terminaux. *Voir l'Aide LabVIEW.*
 - affichage. *Voir l'Aide LabVIEW.*
 - création, 15-3
 - 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.*
- installeurs
 - construction, 7-16
- instances de sous-VIs
 - détermination, 6-11
 - suspension de l'exécution, 6-10
- instances de VIs polymorphes
 - Voir aussi VIs polymorphes.*
 - ajout. *Voir l'Aide LabVIEW.*
 - sélection manuelle, 5-18
 - suppression. *Voir l'Aide LabVIEW.*
- instruments
 - configuration, 1-5
 - contrôle, 7-4
- instruments virtuels. *Voir VIs.*
- intégration d'objets à l'aide d'ActiveX, 19-9
- interface utilisateur. *Voir face-avant.*
- interfaces automation personnalisées, 19-12
- Internet. *Voir Web.*
- ISO Latin-1
 - utilisation du jeu de caractères, 18-19
- IVI
 - drivers d'instrument. *Voir l'Aide LabVIEW.*
 - transmission de noms logiques, 4-23
- J**
 - jauges
 - Voir aussi numériques.*
 - ajout de rampes de couleurs, 4-14
 - face-avant, 4-12
 - jeu de caractères
 - ASCII, 18-19
 - ISO Latin-1, 18-19
 - Mac OS, 18-20
 - utilisation dans les e-mails, 18-18
- L**
 - LabVIEW, 1-1
 - options, 3-9
 - personnalisation, 3-9
 - labview.ini, 3-9
 - lancement de VIs à partir de la ligne de commande. *Voir l'Aide LabVIEW.*
 - lecture de fichiers, 14-1
 - liaison de VIs vers des fichiers d'aide HTML ou compilés. *Voir l'Aide LabVIEW.*
 - liaison vers le fichier journal, 14-16
 - changer, 14-19
 - effacement, 14-18
 - libération de la mémoire. *Voir l'Aide LabVIEW.*
 - licences pour servir des faces-avant distantes, 18-14
 - ligne de commande
 - lancement de VIs. *Voir l'Aide LabVIEW.*
 - lignes de transmission, 13-3
 - liste de paramètres. *Voir connecteurs.*
 - liste déroulante multi-colonnes. *Voir commandes de type liste déroulante.*

liste. *Voir* affichage.
listes déroulantes, 4-17
localisation des VIs, 7-16
login automatique. *Voir l'Aide LabVIEW.*

M

Mac OS

utilisation du jeu de caractères, 18-20

manuel. *Voir* documentation.

mappage

caractères, 18-20

marquage des VIs comme des VIs de niveau principal dans des bibliothèques. *Voir l'Aide LabVIEW.*

masquage

barre de menus, 4-30

barres de défilement, 4-30

éléments optionnels des objets de la face-avant, 4-2

objets de la face-avant. *Voir l'Aide LabVIEW.*

masquage des données numériques, 12-19

masquage des données numériques. *Voir* données numériques

acquisition d'un sous-ensemble, 4-26

mathématiques. *Voir* équations.

MATLAB

nœud de script, 21-5

scripts de mise au point, 21-6

types de données, 21-5

Measurement & Automation Explorer, 1-5

mémoire

compactage. *Voir l'Aide LabVIEW.*

désactivation des outils de mise au point, 6-12

gestion avec un modèle de programmation par flux de données, 5-32

lecture et écriture avec des données variant, 5-26

libération. *Voir l'Aide LabVIEW.*

points de coercition, 5-17

variables globales, 11-6

variables locales, 11-6

menus, 3-4

abrégés, 3-4

chaînes déroulantes, 4-15

commandes de type menu déroulant, 4-19

directs. *Voir l'Aide LabVIEW.*

édition, 16-2

gestion des sélections, 16-3

locaux, 3-4

édition pour les VIs polymorphes.

Voir l'Aide LabVIEW.

référence. *Voir l'Aide LabVIEW.*

menus abrégés, 3-4

menus complets, 3-4

menus déroulants sur la face-avant, 4-19

menus directs. *Voir l'Aide LabVIEW.*

menus locaux

en mode exécution, 3-4

menus raccourcis, 3-4

menus simples, 3-4

message de login au démarrage

affichage. *Voir l'Aide LabVIEW.*

méthodes

ActiveX, 19-6

mise à jour des palettes. *Voir l'Aide LabVIEW.*

mise à jour des VIs, 7-15

mise à l'échelle

graphes, 12-5

objets de la face-avant, 4-8

mise au point

boucles, 6-13

création de sondes. *Voir l'Aide LabVIEW.*

désactivation des outils de mise au point, 6-12

données indéfinies, 6-12

données par défaut, 6-13

fils de liaison masqués, 5-33

gestion automatique d'erreur, 6-14

- options, 3-9
 - outils
 - désactivation, 6-12
 - scripts MATLAB, 21-6
 - sondes, 6-7
 - création. *Voir l'Aide LabVIEW.*
 - structures. *Voir l'Aide LabVIEW.*
 - techniques, 6-4
 - animation de l'exécution, 6-6
 - commentaire excluant une section particulière du diagramme, 6-11
 - gestion d'erreur, 6-14
 - mode pas à pas, 6-6
 - outil Point d'arrêt, 6-9
 - outil Sonde, 6-7
 - suspension de l'exécution, 6-10
 - VIs brisés, 6-2
 - VIs exécutables. *Voir l'Aide LabVIEW.*
 - mise en réseau. *Voir communication.*
 - mises en garde
 - affichage, 6-3
 - affichage par défaut. *Voir l'Aide LabVIEW.*
 - bouton, 6-3
 - mode de conception
 - Conteneur ActiveX, 19-9
 - mode exécution
 - ouverture des VIs dans. *Voir l'Aide LabVIEW.*
 - mode pas à pas
 - mise au point des VIs, 6-6
 - modèle de programmation par flux de données, 5-29
 - gestion de mémoire, 5-32
 - modèle de programmation séquentielle, 5-29
 - modèles
 - création. *Voir l'Aide LabVIEW.*
 - utilisation. *Voir l'Aide LabVIEW.*
 - modules
 - dans les palettes, 3-8
 - motifs
 - terminal, 7-7
- ## N
- NaN (not a number) valeur flottante
 - données indéfinies, 6-12
 - National Instruments
 - filiales internationales, D-2
 - formation pour les clients, D-1
 - services d'intégrateurs système, D-1
 - services professionnels, D-1
 - support technique, D-1
 - .NET
 - assemblées, 19-2
 - Bibliothèques de classes, 19-2
 - Common Language Runtime, 19-2
 - créer un objet. *Voir l'Aide LabVIEW.*
 - environnement, 19-2
 - fonctions., 19-3
 - Global Assembly Cache, 19-3
 - LabVIEW comme client, 19-4
 - mappage des types de données, 19-5
 - Nœud du constructeur, 19-3
 - niveau de coopération
 - paramétrage. *Voir l'Aide LabVIEW.*
 - Nœud d'appel par référence, 17-8
 - Nœud de méthode, 17-5
 - ActiveX, 19-7
 - Nœud de propriété, 17-4
 - ActiveX, 19-11
 - modification des éléments de la liste déroulante, 4-17
 - recherche d'objets ou de terminaux. *Voir l'Aide LabVIEW.*
 - Nœud de rétroaction, 8-10
 - initialisation, 8-11
 - remplacement par des registres à décalage, 8-12
 - nœuds, 2-4
 - appel par référence, 17-8

- appeler, 17-5
- diagramme, 5-7
- flux d'exécution, 5-30
- nœud de script MATLAB, 21-5
- Propriété, 17-4
- redimensionnement. *Voir l'Aide LabVIEW.*
- nœuds d'expression, 21-4
- nœuds de script
 - MATLAB, 21-5
- nœuds extensibles, 7-10
- nombres
 - dépassement des limites inférieures et supérieures, 6-12
- nombres à virgule flottante
 - conversion, B-1
 - dépassement des limites inférieures et supérieures, 6-12
- nombres hors gamme, 4-21
- not a number (NaN) valeur flottante
 - données indéfinies, 6-12
- notes d'application, 1-4
- notification d'erreurs. *Voir l'Aide LabVIEW.*
- numériques
 - chaînes et, 10-5
 - changement de la représentation. *Voir l'Aide LabVIEW.*
 - commandes et indicateurs, 4-11
 - utilisation. *Voir l'Aide LabVIEW.*
 - comparaison, C-2
 - constantes universelles, 5-6
 - conversion, B-1
 - écriture de données sur des fichiers texte ou tableur, 10-5
 - équations, 21-1
 - fonctions, 5-8
 - formatage, 4-12
 - formules, 21-1
 - hors gamme, 4-21
 - polymorphisme, B-2
 - types de données (table), 5-3

- unités de mesures, 5-27
- numéro de révision
 - affichage dans la barre de titre. *Voir l'Aide LabVIEW.*

0

- Objet d'application
 - manipulation des paramètres, 17-4
 - VI Serveur, 17-3
- Objet du VI
 - manipulation des paramètres, 17-4
 - VI Serveur, 17-3
- objets
 - ActiveX, 19-6
 - affichage des éléments optionnels, 4-2
 - alignement, 4-6
 - Voir aussi l'Aide LabVIEW.*
 - câblage automatique sur le diagramme, 5-14
 - câblage manuel sur le diagramme, 5-12
 - changement de commandes en indicateurs et vice versa, 4-2
 - chevauchement sur la face-avant, 4-22
 - coloriage sur la face-avant, 4-5
 - copie des couleurs. *Voir l'Aide LabVIEW.*
 - contrôle par programmation, 17-9
 - création d'info-bulles, 15-3
 - création de descriptions, 15-3
 - déplacement. *Voir l'Aide LabVIEW.*
 - diagramme, 5-1
 - espacement régulier, 4-6
 - Voir aussi l'Aide LabVIEW.*
 - étiquetage, 4-30
 - création. *Voir l'Aide LabVIEW.*
 - édition. *Voir l'Aide LabVIEW.*
 - redimensionnement. *Voir l'Aide LabVIEW.*
 - face-avant et terminaux du diagramme, 5-1

- grouper et verrouiller sur la face-avant, 4-7
 - impression des descriptions, 15-3
 - insertion dans des palettes. *Voir l'Aide LabVIEW.*
 - insertion sur la face-avant à l'aide d'ActiveX, 19-9
 - 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, 4-8
 - paramétrage de l'ordre de défilement par tabulation sur la face-avant, 4-5
 - recherche. *Voir l'Aide LabVIEW.*
 - redimensionnement sur la face-avant, 4-7
 - par rapport à la taille de la fenêtre, 4-8
 - remplacement sur la face-avant, 4-3
 - remplacement sur le diagramme. *Voir l'Aide LabVIEW.*
 - réorganisation. *Voir l'Aide LabVIEW.*
 - répartition, 4-6
 - Voir aussi l'Aide LabVIEW.*
 - sélection. *Voir l'Aide LabVIEW.*
 - sous-titres sur la face-avant, 4-31
 - création. *Voir l'Aide LabVIEW.*
 - transparentes. *Voir l'Aide LabVIEW.*
 - objets masqués de la face-avant. *Voir l'Aide LabVIEW.*
 - OLE pour le protocole DataSocket de contrôle de procédé, 18-4
 - opérateurs. *Voir nœuds.*
 - options
 - enregistrement, 3-9
 - paramétrage, 3-9
 - Voir aussi l'Aide LabVIEW.*
 - options de l'environnement de travail
 - enregistrement, 3-9
 - paramétrage, 3-9
 - Voir aussi l'Aide LabVIEW.*
 - ordre d'exécution, 5-29
 - contrôle avec des structures Séquence, 8-17
 - ordre de navigation. *Voir ordre de défilement par tabulation.*
 - ordre de tabulation
 - paramétrage, 4-5
 - ordre des éléments de cluster, 10-16
 - modification, 10-16
 - Voir aussi l'Aide LabVIEW.*
 - oscillographe, 12-7
 - Outil de publication pour le Web, 18-12
 - outil Point d'arrêt
 - mise au point des VIs, 6-9
 - mise en surbrillance des points d'arrêt. *Voir l'Aide LabVIEW.*
 - outil Sonde
 - mise au point des VIs, 6-7
 - Voir sondes.*
 - outils
 - palette, 3-3
 - sélection manuelle. *Voir l'Aide LabVIEW.*
 - ouverture des VIs en mode exécution. *Voir l'Aide LabVIEW.*
- ## P
- palettes
 - Commandes, 3-1
 - personnalisation, 3-6
 - Fonctions, 3-2
 - personnalisation, 3-6
 - insertion d'objets. *Voir l'Aide LabVIEW.*
 - mise à jour. *Voir l'Aide LabVIEW.*
 - modifications. *Voir l'Aide LabVIEW.*
 - modules, 3-8
 - navigation et recherche, 3-2
 - options, 3-9
 - organisation, 3-7

- Outils, 3-3
 - partage. *Voir l'Aide LabVIEW.*
 - personnalisation, 3-6
 - référence. *Voir l'Aide LabVIEW.*
 - sélecteur de couleurs, 4-13
 - toolsets, 3-8
 - types, 3-8
- panneau arrière. *Voir diagramme.*
- paramétrage
 - niveau de coopération. *Voir l'Aide LabVIEW.*
 - options de l'environnement de travail, 3-9
 - Voir aussi l'Aide LabVIEW.*
- paramètre de chaîne de format, 10-4
- paramètres
 - types de données (table), 5-3
- paramètres dupliqués, 14-13
- partage
 - des fichiers, 7-2
 - données en direct avec d'autres VIs et applications, 18-2
 - données en direct par programmation, 18-7
 - types de palettes. *Voir l'Aide LabVIEW.*
- VIs, 7-15
- partage de fichiers, 7-2
- pas à pas dans les VIs
 - mise au point des VIs, 6-6
- Pas un chemin, 4-16
- performance
 - désactivation des outils de mise au point, 6-12
 - options, 3-9
 - variables globales et locales, 11-5
- personnalisation
 - apparence et comportement des VIs, 16-1
 - codes d'erreur. *Voir l'Aide LabVIEW.*
 - environnement de travail, 3-6
 - menus, 16-2
 - palettes, 3-6
 - sondes. *Voir l'Aide LabVIEW.*
- pires
 - données variant, 5-26
- planification des projets, 7-1
- planification du projet, 7-1
- plusieurs threads
 - exécution. *Voir l'Aide LabVIEW.*
- points de coercition, 5-17
- police de l'application, 4-32
- police du système, 4-32
- polices
 - application, 4-32
 - dialogue, 4-32
 - options, 3-9
 - paramètres, 4-31
 - système, 4-32
- polymorphes
 - fonctions, B-1
 - nœuds d'expression, 21-5
 - unités, B-1
 - VIs, 5-18
 - ajout d'instances. *Voir l'Aide LabVIEW.*
 - construction, 5-19
 - édition des menus locaux. *Voir l'Aide LabVIEW.*
 - retrait des VIs. *Voir l'Aide LabVIEW.*
 - sélection manuelle d'une instance. *Voir l'Aide LabVIEW.*
- PPC Toolbox, 18-22
- préférences. *Reportez-vous à options.*
- programmation événementielle *Voir événements; structures Événement.*
- programmes d'exemple, D-1
- propriétés
 - ActiveX, 19-6
 - paramétrage, 19-10
 - par programmation, 19-11
 - visualisation, 19-10
- protection par mot de passe, 7-16
- protocole DataSocket dstp, 18-3
- protocole DataSocket file, 18-4

protocole DataSocket ftp, 18-4
protocole DataSocket logos, 18-4
protocole DataSocket opc, 18-4
protocoles
 communication de bas niveau, 18-21
 DataSocket, 18-3
publication de VIs sur le Web, 18-11

R

raccourcis clavier, 4-4
 contrôler les boutons, 4-4
 définition de l'ordre de défilement par
 tabulation, 4-5
rafraîchissements progressifs
 pendant le traçage. *Voir l'Aide LabVIEW.*
 pour les graphes, 12-6
rapports
 génération, 15-8
 clusters d'erreur. *Voir l'Aide
 LabVIEW.*
 impression, 15-8
 VIs de génération de rapport, 15-8
recherche
 commandes, VIs et fonctions sur les
 palettes, 3-2
 erreurs, 6-2
 hiérarchie du VI. *Voir l'Aide LabVIEW.*
 objets, texte et VIs. *Voir l'Aide LabVIEW.*
 versions PDF de la documentation
 LabVIEW, 1-1
récupération des données
 par programmation, 14-19
 utilisation de sous-VIs, 14-19
 utilisation des fonctions d'E/S sur
 fichier, 14-21
redimensionnement
 clusters. *Voir l'Aide LabVIEW.*
 constantes définies par l'utilisateur, 5-7
 étiquettes. *Voir l'Aide LabVIEW.*
 fonctions. *Voir l'Aide LabVIEW.*

 nœuds. *Voir l'Aide LabVIEW.*
 objets de la face-avant, 4-7
 par rapport à la taille de la fenêtre, 4-8
 sous-VIs, 7-10
 tableaux. *Voir l'Aide LabVIEW.*
 tables. *Voir l'Aide LabVIEW.*
 VIs Express, 7-10
références de commande partiellement
 spécifiées, 17-10
références de commandes, 17-9
 création. *Voir l'Aide LabVIEW.*
 de type strict, 17-10
 partiellement spécifiées, 17-10
refnums
 automation, 19-7
 commande, 17-9
 commandes et indicateurs, 4-29
 type de données (table), 5-4
 utilisation. *Voir l'Aide LabVIEW.*
 de type strict, 17-8
 E/S sur fichiers, 14-1
 Nœud d'appel par référence, 17-8
refnums de type strict
 commande, 17-10
 VI, 17-8
registres à décalage, 8-6
 remplacement par des tunnels, 8-9
remplacement
 des objets de la face-avant, 4-3
 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
répartition
 des objets de la face-avant, 4-6

Voir aussi l'Aide LabVIEW.
 VIs, 7-15
 répartition des objets, 4-6
Voir aussi l'Aide LabVIEW.
 répertoire de structure et support
 menus, A-2
 projet, A-2
 ressource, A-2
 templates, A-2
 WWW, A-2
 répertoire exercices et exemples
 exemples, 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, 12-15
 réservoirs
 Voir aussi numériques.
 commandes et indicateurs à glissières, 4-11
 résolutions écran, 4-35
 ressources de diagnostic, D-1
 ressources pour le dépannage, D-1
 revenir aux dernières versions enregistrées.
 Voir l'Aide LabVIEW.
 routage automatique, 5-15
 routage des fils de liaison, 5-15
 RTF
 enregistrement de la documentation dans un fichier, 15-4

S

saisie semi-automatique, 4-17
 listes déroulantes, 4-17

schéma pour le format XML, 10-8
 sélection
 fils de liaison, 5-16
 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
 activation, 18-11
 clients pour les faces-avant distantes
 LabVIEW, 18-15
 multiples, 18-14
 navigateur Web, 18-15
 contrôle de VIs, 18-13
 licences pour servir des faces-avant distantes, 18-14
 options, 18-11
 visualisation de faces-avant, 18-13
 serveurs
 ActiveX, 19-12
 configuration pour les faces-avant distantes, 18-13
 LabVIEW, 18-15
 navigateur Web, 18-16
 services d'intégrateurs système, D-1
 services professionnels, D-1
 situations de compétition, 11-5
 SMTP
 en utilisant des VIs, 18-18
 jeux de caractères, 18-18
 translitération, 18-20
 sondes
 création. *Voir l'Aide LabVIEW.*
 fournies, 6-8
 Générique, 6-7
 indicateurs, 6-8

- mise au point des VIs, 6-7
- personnalisées, 6-9
 - création. *Voir l'Aide LabVIEW.*
- sondes par défaut, 6-8
- types de, 6-7
- sondes avancées. *Voir sondes.*
- sondes fournies, 6-8
- sondes intelligentes. *Voir sondes.*
- sondes par défaut, 6-8
- sondes utilisateur. *Voir sondes.*
- sous-palettes
 - construction d' ActiveX, 3-8
 - création. *Voir l'Aide LabVIEW.*
 - déplacement. *Voir l'Aide LabVIEW.*
 - organisation, 3-7
- sous-programme. *Voir sous-VIs.*
- sous-titres, 4-31
 - création. *Voir l'Aide LabVIEW.*
 - info-bulles des sous-VIs. *Voir l'Aide LabVIEW.*
- sous-VIs
 - affichage de la chaîne d'appelants, 6-11
 - affichage des noms lorsque placés. *Voir l'Aide LabVIEW.*
 - conception, 7-11
 - construction, 7-5
 - contrôle du comportement, 7-4
 - copie, A-3
 - création, 7-11
 - situations à éviter. *Voir l'Aide LabVIEW.*
 - création à partir de VIs Express, 5-22
 - Voir aussi l'Aide LabVIEW.*
 - détermination de l'instance actuelle, 6-11
 - face-avant, 7-11
 - faces-avant distantes, 18-16
 - hiérarchie, 7-12
 - icônes, 7-10
 - impression de données à partir d'un VI de plus haut niveau, 15-8
 - nœuds extensibles, 7-10
 - récupération des données de la face-avant, 14-19
 - sous-titres des commandes pour les info-bulles. *Voir l'Aide LabVIEW.*
 - suspension de l'exécution, 6-10
 - VIs polymorphes, 5-18
- structure des répertoires de LabVIEW, A-1
 - Mac OS, A-2
 - organisation, A-1
- structures, 8-1
 - boucles For, 8-2
 - boucles While, 8-3
 - câblage *Voir l'Aide LabVIEW.*
 - Condition, 8-13
 - Événement, 9-3
 - mise au point. *Voir l'Aide LabVIEW.*
 - Séquence déroulée, 8-16
 - Séquence empilée, 8-16
 - suppression. *Voir l'Aide LabVIEW.*
 - sur le diagramme, 2-4
 - utilisation. *Voir l'Aide LabVIEW.*
 - variables globales, 11-2
 - variables locales, 11-1
- structures Condition
 - gestion d'erreur, 6-17
 - spécification d'une condition par défaut, 8-13
 - terminaux de sélecteurs
 - valeurs, 8-13
 - types de données, 8-13
 - utilisation. *Voir l'Aide LabVIEW.*
- structures Événement
 - Voir aussi événements.*
 - utilisation, 9-3
- structures Séquence
 - Voir Structures Séquence déroulée; Structures Séquence empilée*
 - Accès aux valeurs avec des variables globales et locales, 11-4
 - comparaison entre la séquence empilée et la séquence déroulée, 8-15

- contrôle de l'ordre d'exécution, 5-30
- usage excessif, 8-17
- utilisation. *Voir l'Aide LabVIEW.*
- structures Séquence déroulée, 8-16
 - Voir aussi* structures Séquence.
 - remplacement par une Séquence empilée, 8-19
- Structures Séquence empilée, 8-16
 - Voir aussi* structures Séquence.
 - remplacement par une Séquence déroulée, 8-19
- style des commandes et des indicateurs, 16-2
- suivi du développement. *Reportez-vous à* documentation des VIs.
- support technique, D-1
- support technique en ligne, D-1
- support technique international, D-2
- support technique par téléphone, D-2
- suppression
 - des enregistrements de données, 14-18
 - éléments de tableau. *Voir l'Aide LabVIEW.*
 - fils de liaison brisés, 5-16
 - instances de VIs polymorphes. *Voir l'Aide LabVIEW.*
 - objets sur la face-avant ou le diagramme. *Voir l'Aide LabVIEW.*
 - structures. *Voir l'Aide LabVIEW.*
 - terminaux provenant des fonctions, 5-12
 - types de palettes. *Voir l'Aide LabVIEW.*
 - VIs des bibliothèques. *Voir l'Aide LabVIEW.*
- suspension de l'exécution
 - mise au point des VIs, 6-10
- synchronisation
 - contrôle, 8-12

T

- tableaux
 - auto-indexation des boucles, 8-4

- commandes et indicateurs, 4-17
 - type de données (table), 5-4
- comparaison, C-2
- constantes, 10-12
- construction au moyen de boucles, 8-6
- conversion des clusters. *Voir l'Aide LabVIEW.*
- création, 10-12
- déplacement. *Voir l'Aide LabVIEW.*
- dimensions, 10-9
- données par défaut, 6-13
- exemples
 - tableaux 1D, 10-9
 - tableaux 2D, 10-10
- fonctions, 5-9
- indices, 10-9
 - affichage, 10-13
- insertion d'éléments. *Voir l'Aide LabVIEW.*
- polymorphisme, B-5
- redimensionnement. *Voir l'Aide LabVIEW.*
- remplacement des éléments. *Voir l'Aide LabVIEW.*
- restrictions, 10-12
- suppression d'éléments. *Voir l'Aide LabVIEW.*
- taille des, 6-13
 - variables globales, 11-6
- tables, 4-19, 10-3
 - utilisation. *Voir l'Aide LabVIEW.*
- tables de pixels, 13-4
- taille de la fenêtre
 - définition. *Voir l'Aide LabVIEW.*
- TCP, 18-21
 - VI Serveur, 17-1
- terminaux, 2-3
 - affichage, 5-2
 - affichage des info-bulles. *Voir l'Aide LabVIEW.*
 - ajout de fonctions, 5-12

- apparence dans la fenêtre Aide
 - contextuelle, 7-9
- câblage, 5-12
- commandes et indicateurs (table), 5-3
- conditionnels, 8-3
- constantes, 5-5
- décompte, 8-2
 - auto-indexation pour définir, 8-5
- diagramme, 5-2
- impression, 15-3
- itération
 - boucles For, 8-2
 - boucles While, 8-3
- motifs, 7-7
- nécessaires, 7-8
- objets de la face-avant et, 5-1
- optionnels, 7-8
- points de coercition, 5-17
- recherche. *Voir l'Aide LabVIEW.*
- recommandés, 7-8
- sélecteur, 8-13
- suppressions en provenance de fonctions, 5-12
- variable locale de séquence, 8-17
- 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, 8-13
- terminaux de variable locale de séquence, 8-17
- terminaux récepteurs. *Voir indicateurs.*
- terminaux source. *Voir commandes.*
- texte
 - commandes de type menu déroulant, 4-19
 - faire glisser et déposer. *Voir l'Aide LabVIEW.*
 - formatage, 4-31
 - recherche. *Voir l'Aide LabVIEW.*
 - zones de saisie, 4-14
- thermomètres
 - Voir aussi* numériques.
 - commandes et indicateurs à glissières, 4-11
- 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, 3-8
- toolsets supplémentaires, 1-1
 - dans les palettes, 3-8
- touches de fonction par défaut
 - ignorer par défaut. *Voir l'Aide LabVIEW.*
- traçage
 - Voir aussi* graphiques.
 - rafraîchissements progressifs. *Voir l'Aide LabVIEW.*
- tracé de lignes
 - lissage, 12-2
- transfert des VIs, 7-16
- translitération
 - envoi d'e-mails, 18-20
- Transmission Control Protocol, 18-21
- transparent
 - étiquettes. *Voir l'Aide LabVIEW.*
 - objets. *Voir l'Aide LabVIEW.*
- tunnels, 8-1
 - entrée et sortie, 8-15
 - remplacement par des registres à décalage, 8-9
- type de données dynamiques, 5-22
 - conversion à partir du, 5-24
 - Voir aussi l'Aide LabVIEW.*
 - conversion en, 5-25
 - Voir aussi l'Aide LabVIEW.*
- type de données waveform numérique, 12-21
- types, 3-8
 - création, 3-7

- édition, 3-7
- modifications. *Voir l'Aide LabVIEW.*
- partage. *Voir l'Aide LabVIEW.*
- suppression. *Voir l'Aide LabVIEW.*
- types de données
 - commandes et indicateurs (table), 5-3
 - conversion à partir d'XML, 10-8
 - conversion en XML, 10-7
 - données waveform, 12-20
 - impression, 15-3
 - MATLAB (table), 21-5
 - .NET, 19-5
 - structures Condition, 8-13
 - valeurs par défaut, 5-3

U

- UDP, 18-21
- unités de mesures, 5-27
- URL pour DataSocket, 18-3
- User Datagram Protocol, 18-21
- utilisateur
 - enregistrement utilisateur, 9-15
- utilisation de chaînes de formatage. *Voir l'Aide LabVIEW.*
- utilisation de la touche tab pour naviguer entre les objets de la face-avant, 4-5
- Utilitaire de configuration NI-DAQ, 1-5

V

- valeurs par défaut
 - types de données, 5-3
- variables
 - globales
 - création, 11-3
 - initialisation, 11-5
 - lecture et écriture, 11-4
 - mémoire, 11-6
 - situations de compétition, 11-5

- utilisation mesurée, 11-4
 - locales, 11-1
 - création, 11-2
 - initialisation, 11-5
 - lecture et écriture, 11-4
 - mémoire, 11-6
 - situations de compétition, 11-5
 - utilisation mesurée, 11-4
 - variables globales
 - création, 11-3
 - initialisation, 11-5
 - lecture et écriture, 11-4
 - mémoire, 11-6
 - situations de compétition, 11-5
 - utilisation mesurée, 11-4
 - variables globales d'écriture, 11-4
 - variables globales de lecture, 11-4
 - variables locales, 11-1
 - création, 11-2
 - initialisation, 11-5
 - lecture et écriture, 11-4
 - mémoire, 11-6
 - recherche d'objets ou de terminaux. *Voir l'Aide LabVIEW.*
 - situations de compétition, 11-5
 - utilisation mesurée, 11-4
 - variables locales d'écriture, 11-4
 - variables locales de lecture, 11-4
 - vérification de l'espace disque disponible. *Voir l'Aide LabVIEW.*
 - vérification de type stricte, 5-27
 - verrouillage
 - face-avant avec des événements, 9-8
 - fenêtre Aide contextuelle, 3-5
 - objets de la face-avant, 4-7
 - VI. *Voir l'Aide LabVIEW.*
 - versions
 - comparaison, 7-2
 - enregistrement des VIs pour la précédente version, 7-15

- revenir aux dernières enregistrées. *Voir l'Aide LabVIEW.*
- versions antérieures
 - enregistrement des VIs, 7-15
- VI Callback
 - ActiveX, 19-15
- VI Serveur
 - appel à distance des VIs, 17-1
 - appel d'autres instances de LabVIEW sur le Web, 17-1
 - applications à distance, 17-9
 - caractéristiques, 17-1
 - construire des applications, 17-2
 - contrôle des objets de la face-avant, 17-9
 - manipulation des paramètres de VI, 17-4
 - mise en réseau et, 18-1
 - Nœud d'appel par référence, 17-8
 - Nœud de méthode, 17-5
 - Nœud de propriété, 17-4
 - Objet d'application, 17-3
 - Objet du VI, 17-3
 - refnums de VI de type strict, 17-8
- VI System Exec, 18-22
- VIs, 2-1
 - ajouter à des bibliothèques, 3-7
 - appel à distance, 17-1
 - appeler de façon dynamique, 17-8
 - appellation, 7-15
 - bibliothèques, 7-13
 - brisés, 6-2
 - charger de façon dynamique, 17-8
 - comparer des versions, 7-2
 - configuration de l'apparence et du comportement, 16-1
 - construction, 7-1
 - contrôle lorsqu'ils sont appelés comme sous-VIs, 7-4
 - contrôle par programmation, 17-1
 - contrôle sur le Web, 18-11
 - copie, A-3
 - correction, 6-2
 - création de descriptions, 15-3
 - développement, 7-1
 - déverrouillage. *Voir l'Aide LabVIEW.*
 - documentation, 15-1
 - enregistrement, 7-13
 - exécutables
 - mise au point. *Voir l'Aide LabVIEW.*
 - exécution, 6-1
 - exemples, 1-5
 - faire glisser et déposer. *Voir l'Aide LabVIEW.*
 - gestion d'erreur, 6-14
 - hiérarchie, 7-12
 - impression, 15-6
 - lancement de la ligne de commande. *Voir l'Aide LabVIEW.*
 - localisation., 7-16
 - marquage comme étant de niveau principal dans des bibliothèques. *Voir l'Aide LabVIEW.*
 - mise à jour, 7-15
 - ouverture en mode exécution. *Voir l'Aide LabVIEW.*
 - partage, 7-15
 - polymorphes, 5-18
 - publication sur le Web, 18-11
 - recherche. *Voir l'Aide LabVIEW.*
 - référence. *Voir l'Aide LabVIEW.*
 - refnums de type strict, 17-8
 - répartition, 7-15
 - retrait des bibliothèques. *Voir l'Aide LabVIEW.*
 - revenir aux dernières versions enregistrées. *Voir l'Aide LabVIEW.*
 - techniques de mise au point, 6-4
 - transfert, 7-16
 - 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, 14-15
 lecture et écriture de fichiers .ini, 14-13
 spécial, 14-14
 VIs exécutables
 mise au point. *Voir l'Aide LabVIEW.*
 VIs Express, 5-22
 enregistrement de leurs configurations en
 tant que VIs, 5-22
 Voir aussi l'Aide LabVIEW.
 icônes, 7-10
 nœuds extensibles, 7-10
 VISA
 transmission de noms de ressource, 4-23
 visualisation *Voir* affichage.
 vitesse d'exécution
 contrôle, 8-12
 vitesse de clignotement. *Voir l'Aide LabVIEW.*
 voyants sur la face-avant, 4-14
 VXI
 configuration, 1-5
 VIs, 1-4

W

waveforms
 écrire dans des fichiers, 14-11
 lecture de fichiers, 14-12
 Web
 appel d'autres instances de
 LabVIEW, 17-1
 contrôle de VIs, 18-13
 création de documents HTML, 18-12
 publication de VIs, 18-11
 services professionnels, D-1
 support technique, D-1
 visualisation de faces-avant, 18-13

X

XML
 conversion à partir de, 10-8
 Conversion de types de données en, 10-7
 exemple, 10-6
 schéma, 10-8

Z

zoom sur des graphes et graphes déroulants.
Voir l'Aide LabVIEW.